

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES



Universidad
Politécnica
de Cartagena



E.T.S.I.T

Nuevas técnicas basadas en Redes Neuronales para el
diseño de filtros de microondas multicapa apantallados.

(Tesis doctoral)

28 de octubre de 2009

AUTORES:

Doctorando: Juan Pascual García

Directores: Alejandro Álvarez Melcón

Fernando Quesada Pereira

Esta tesis doctoral ha sido financiada
a través de una beca de Formación del Personal Investigador (FPI)
por la Fundación Séneca, Agencia de Ciencia y Tecnología de la Región de Murcia



UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Comisión de Doctorado

**AUTORIZACIÓN DE LA PRESENTACIÓN DE LA TESIS DOCTORAL
POR LA COMISIÓN ACADÉMICA RESPONSABLE DEL PROGRAMA**

D./D^a. Pedro Sánchez Palma, Coordinador de la Comisión Académica del
Programa Tecnologías de la Información y Comunicaciones

INFORMA:

Que la Tesis Doctoral titulada "Nuevas técnicas basadas en Redes Neuronales para el diseño de filtros de microondas multicapa apantallados", ha sido realizada por D./D^a. Juan Pascual García, bajo la dirección y supervisión de D./D^a. Alejandro Álvarez Melcón y Fernando Quesada Pereira y que la Comisión Académica ha dado su conformidad para que sea presentada ante la Comisión de Doctorado.

En Cartagena, a 26 de octubre de 2009

EL COORDINADOR

PROGRAMA

Fdo.: Pedro Sánchez Palma





AUTORIZACIÓN DE LA PRESENTACIÓN DE LA TESIS DOCTORAL
POR EL DIRECTOR/A

D/D^a. Alejandro Álvarez Melcón, Profesor/a Doctor/a del Área de Teoría de la Señal y Comunicaciones en el Departamento de _Tecnologías de la Información y Comunicaciones

AUTORIZA:

La presentación de la Tesis Doctoral titulada "Nuevas técnicas basadas en Redes Neuronales para el diseño de filtros de microondas multicapa apantallados", realizada por D./D^a. Juan Pascual García, bajo mi dirección y supervisión, en el Departamento de Tecnologías de la Información y Comunicaciones, y que presenta para la obtención del grado de Doctor por la Universidad Politécnica de Cartagena.

En Cartagena, a 26 de OCTUBRE de 2008

EL/LA DIRECTOR/A DE TESIS





AUTORIZACIÓN DE LA PRESENTACIÓN DE LA TESIS DOCTORAL
POR EL DIRECTOR/A

D/D^a. Fernando Quesada Pereira, Profesor/a Doctor/a del Área de Teoría de la Señal y Comunicaciones en el Departamento de Tecnologías de la Información y Comunicaciones

A U T O R I Z A:

La presentación de la Tesis Doctoral titulada "Nuevas técnicas basadas en Redes Neuronales para el diseño de filtros de microondas multicapa apantallados", realizada por D./D^a. Juan Pascual García, bajo mi dirección y supervisión, en el Departamento de Tecnologías de la Información y Comunicaciones, y que presenta para la obtención del grado de Doctor por la Universidad Politécnica de Cartagena.

En Cartagena, a 26 de OCTUBRE de 2009

EL/LA DIRECTOR/A DE TESIS

Fdo.:

Agradecimientos

En cualquier actividad que realice tengo agradecer, siempre en primer lugar, a mi madre y a mi padre, a mis dos hermanas todo el amor y cariño que tienen hacia mí. Sin ellos todo lo que he llevado a cabo hubiera sido mucho más difícil. Siempre me han apoyado y aconsejado cuando era necesario. En todas las tesis doctorales se debe dar las gracias al director del trabajo científico por su ayuda y consejo. En el caso de mi director, Alejandro, no sólo ha solucionado mis dudas, indicado posibles mejoras o corregido el trabajo que se llevaba a cabo, sino que su cercanía y carácter han conseguido que todas las tareas que se debían realizar en general, y la tesis doctoral en particular, hayan sido de agradable cumplimiento. Su dedicación y esfuerzo a la dirección de esta tesis doctoral han sido encomiables. También quiero agradecer el apoyo, la ayuda y el afecto a los miembros del grupo de investigación donde he desarrollado mi trabajo, el GEAT. Primero, a los que encontré cuando vine a trabajar a la UPCT. David, al que conocí en la carrera universitaria en Valencia, y que fue el amigo que me indicó la posibilidad de trabajar en Cartagena. Fernando y José Luis, cuya amistad he ido cultivando desde que vine a Cartagena. También a Pedro Vera al que conocí cuando entré en el GEAT. En segundo lugar, tengo un recuerdo cordial para todos aquellos que se han ido incorporando al grupo después de mí, Javi, Sebas, Mónica. No me puedo olvidar de todos mis amigos y compañeros del área y del departamento. Con ellos he pasado muy buenos momentos y me ayudado cuando he solicitado su colaboración. También tengo un recuerdo cariñoso para mis compañeros del primer grupo de investigación, el CEMI, en el que trabajé cuando comencé mi andadura en la UPCT. Quiero agradecer a la Fundación Séneca la financiación de esta tesis doctoral gracias a la beca FPI que me concedieron. Además, siempre han apoyado mediante ayudas, la divulgación en congresos científicos de los trabajos realizados. Por último, agradecer a Alejandro y a Fernando, que ha sido codirector de la tesis, todas las correcciones efectuadas en este manuscrito.

Resumen

En la presente tesis se ha desarrollado un método de análisis de circuitos apantallados multicapa basado en redes neuronales. Una de las técnicas más utilizadas y exitosas en el análisis de este tipo de circuitos es la Ecuación Integral (IE) resuelta mediante el Método de los Momentos (MoM). La mayor carga computacional en la resolución de la IE proviene de la evaluación numérica de las funciones de Green de los potenciales relevantes del medio. En este trabajo se ha conseguido acelerar drásticamente el análisis circuital gracias a la aproximación de las funciones de Green mediante redes neuronales. Una vez entrenadas, las redes neuronales sustituyen a las funciones de Green en la IE. Se han utilizado dos tipos de redes neuronales: las Redes neuronales de funciones de base radial (RBFNN) y las Redes neuronales de Chebychev. Gracias principalmente a dos operaciones ha sido posible la correcta aproximación de las funciones de Green. Por un lado, se ha desarrollado un método de división del espacio de entrada muy efectivo. Por otro lado, la eliminación de la singularidad posibilita la aproximación de funciones de variación lenta. Se han aplicado dos estrategias de eliminación de la singularidad de las funciones de Green. La primera está basada en la multiplicación por la distancia (ρ) entre punto fuente y observación. La segunda técnica supera con creces a la anterior y consiste en la extracción de dos capas de imágenes espaciales del sumatorio total de imágenes. Respecto a las Redes neuronales de Chebychev se ha aplicado de modo novedoso el algoritmo de entrenamiento OLS. Este método permite el diseño óptimo de este tipo de redes, de modo que el rendimiento de estas redes ha superado en gran medida al de las RBFNNs. En ambas redes neuronales, la ganancia conseguida en el tiempo de análisis convierte en rentable al método neuronal. El tiempo invertido en la división del espacio de entrada y en el entrenamiento del conjunto de redes neuronales llega a ser despreciable con tan solo unos pocos análisis circuitales.

Como demostración práctica de la habilidad de los métodos neuronales de análisis programados se han desarrollado dos nuevos procedimientos de diseño de filtros encapsulados. El primer método utiliza los Algoritmos genéticos para optimizar un filtro inicial que no cumple las especificaciones prefijadas. Con el objetivo de asegurar la convergencia del proceso de optimización se ha definido una nueva función de rendimiento especialmente adecuada para el diseño de filtros. Esta nueva función mide el grado de cumplimiento de las especificaciones e impide la aparición del problema de la convergencia prematura. El segundo método se basa en la aproximación con redes neuronales de las relaciones entre los parámetros eléctricos, que definen la respuesta de un filtro, y las dimensiones físicas que sintetizan dichos parámetros. Las redes neuronales entrenadas con estos datos pueden ser usadas en el diseño de muchos filtros en una estructura dada. Ambos métodos han demostrado su validez en el diseño de filtros prácticos.

Abstract

In this PhD thesis one method of shielded multilayer circuit neural network based analysis has been developed. One of the most successful analysis procedures of these kind of structures is the Integral Equation technique (IE) solved by the Method of Moments (MoM). In order to solve the IE, in the version which uses the media relevant potentials, it is necessary to have a formulation of the Green's functions associated to the mentioned potentials. The main computational burden in the IE resolution lies on the numerical evaluation of the Green's functions. In this work, the circuit analysis has been drastically accelerated thanks to the approximation of the Green's functions by means of neural networks. Once trained, the neural networks substitute the Green's functions in the IE. Two different types of neural networks have been used: the Radial basis function neural networks (RBFNN) and the Chebyshev neural networks. Thanks mainly to two distinct operations the correct approximation of the Green's functions has been possible. On the one hand, a very effective input space division has been developed. On the other hand, the elimination of the singularity makes feasible the approximation of slow variation functions. Two different singularity elimination strategies have been developed. The first one is based on the multiplication by the source-observation points distance (ρ). The second one outperforms the first one. It consists of the extraction of two layers of spatial images from the whole summation of images. With regard to the Chebyshev neural networks, the OLS training algorithm has been applied in a novel fashion. This method allows the optimum design in this kind of neural networks. In this way, the performance of these neural networks outperforms greatly the RBFNNs one. In both networks, the time gain reached makes the neural method profitable. The time invested in the input space division and in the neural training is negligible with only few circuit analysis.

To show, in a practical way, the ability of the neural based analysis method, two new design procedures have been developed. The first method uses the Genetic Algorithms to optimize an initial filter which does not fulfill the established specifications. A new fitness function, specially well suited to design filters, has been defined in order to assure the correct convergence of the optimization process. This new function measures the fulfillment of the specifications and it also prevents the appearance of the premature convergence problem. The second method is found on the approximation, by means of neural networks, of the relations between the electrical parameters, which defined the circuit response, and the physical dimensions that synthesize the aforementioned parameters. The neural networks trained with these data can be used in the design of many circuits in a given structure. Both methods have shown their ability in the design of practical filters.

Es lo malo de no hacer imprimir las obras,
que se va la vida en rehacerlas

ALFONSO REYES, *Cuestiones gongorinas*, 60
citado por JORGE LUIS BORGES en *Discusión*

Índice general

1. Introducción	1
1.1. Visión general del trabajo y objetivos de la tesis	1
1.2. Métodos de análisis de estructuras apantalladas de microondas	4
1.3. Solución propuesta para acelerar el análisis de circuitos de microondas encapsulados e impresos	8
1.3.1. Redes neuronales	8
1.3.2. Clases de redes neuronales: visión histórica	11
1.3.3. Metodología	13
1.4. Nuevos métodos de diseño de circuitos de microondas encapsulados y apantallados. .	14
1.4.1. Técnica de diseño basada en Algoritmos genéticos.	16
1.4.2. Técnica de diseño basada en redes neuronales y aproximación de parámetros eléctricos.	17
1.5. Estructura de la Tesis	18
1.6. Contribuciones originales de la presente tesis	20
2. Evaluación de las funciones de Green del medio encapsulado multicapa	23
2.1. Introducción	23
2.2. Formulación de las funciones de Green de los potenciales del medio multicapa apantallado	25
2.3. Método híbrido	30
2.4. Método de suma en el dominio espectral: suma por partes	40

2.5. Propiedades de las funciones de Green	50
2.5.1. Singularidad en la fuente	50
2.5.2. Simetría de la cavidad	52
2.6. Conclusiones	55
3. Método neuronal de análisis de circuitos apantallados multicapa basado en Redes neuronales de funciones de base radial	57
3.1. Introducción	57
3.2. Selección de la red neuronal	61
3.2.1. Perceptrón multicapa (MLP)	61
3.2.2. Máquinas de vectores soporte (SVM)	63
3.2.3. Redes neuronales de funciones de base radial (RBFNN)	65
3.3. Algoritmos de entrenamiento de las Redes neuronales de funciones de base radial . .	71
3.3.1. Selección de centros aleatoria	72
3.3.2. Algoritmo de agrupación en K-medias (K-means)	73
3.3.3. Algoritmo de Mínimos cuadrados ortogonales (OLS)	74
3.4. Extracción de la singularidad	82
3.5. Rendimiento de los algoritmos de entrenamiento en la aproximación de las funciones de Green	85
3.6. Comparativa entre RBFNN, MLP y SVM en la aproximación de las funciones de Green	97
3.7. Algoritmo de división del espacio de entrada. Varianza fija	101
3.8. Algoritmo de división del espacio de entrada con varianza adaptativa	113
3.9. Eliminación de la singularidad basada en la extracción de capas de imágenes espaciales. Zonas de observación circulares	128
3.9.1. Versión mejorada del algoritmo OLS	134
3.9.2. Test del rendimiento de las nuevas técnicas en la aproximación de las funciones de Green	136

3.9.3. Análisis de circuitos encapsulados con las nuevas técnicas de aproximación de las funciones de Green	142
3.10. Conclusiones	162
4. Método neuronal de análisis de circuitos apantallados multicapa basado en Redes neuronales de Chebychev	167
4.1. Introducción	167
4.2. Eficiencia del algoritmo OLS en la selección de las neuronas de una Red neuronal de funciones de base radial	170
4.3. Solución al problema de selección de regresores mediante el algoritmo OLS: Redes neuronales de Chebychev	173
4.4. Redes neuronales de Chebychev: algoritmo de entrenamiento	177
4.4.1. Espacio de entrada unidimensional-espacio de salida unidimensional	177
4.4.2. Espacio de entrada multidimensional-espacio de salida unidimensional	181
4.4.3. Espacio de entrada multidimensional-espacio de salida multidimensional	184
4.5. Capacidad de aproximación de las Redes neuronales de Chebychev	186
4.6. Carga computacional en la selección de los regresores	189
4.7. Tests numéricos de la capacidad de aproximación de las Redes neuronales de Chebychev	191
4.8. Capacidad de las Redes neuronales de Chebychev en la aproximación de las funciones de Green	200
4.8.1. Zonas observación y fuente en las Redes neuronales de Chebychev	200
4.8.2. Aproximación de las funciones de Green	203
4.9. Ventajas y desventajas de las Redes neuronales de Chebychev frente a las Redes neuronales de funciones de base radial	213
4.10. Análisis de circuitos encapsulados con las Redes neuronales de Chebychev	215
4.11. Comparativa con otros métodos de análisis	222
4.12. Análisis de circuitos encapsulados con Redes neuronales de Chebychev en cavidades variables	230
4.13. Conclusiones	236

5. Nuevos métodos de diseño de filtros apantallados multicapa	241
5.1. Introducción	241
5.2. Método de diseño de filtros apantallados multicapa basado en Algoritmos genéticos .	243
5.2.1. Modelo del Algoritmo genético	243
5.2.2. Nueva función de rendimiento	245
5.2.3. Método de diseño de filtros	248
5.2.4. Ejemplos de diseño de filtros apantallados	249
5.3. Método de diseño de filtros apantallados multicapa basado en redes neuronales	261
5.3.1. Introducción	261
5.3.2. Cálculo de las dimensiones del filtro a partir de los parámetros circuitales mediante redes neuronales	263
5.3.3. Método de diseño basado en Redes neuronales de funciones de base radial . .	268
5.3.4. Ejemplos de diseño de filtros apantallados con el método de diseño basado en Redes neuronales de funciones de base radial: simulador electromagnético basado en Redes neuronales de funciones de base radial	274
5.3.5. Ejemplos de diseño de filtros apantallados con el método de diseño basado en Redes neuronales de funciones de base radial: simulador electromagnético basado en Redes neuronales de Chebychev	280
5.4. Conclusiones	294
6. Conclusiones y líneas futuras	297
Apéndices	
A. Series infinitas de funciones sinusoidales	313
B. Ortogonalidad de los regresores en la red neuronal de Chebychev en un espacio de entrada multidimensional	315
C. Aplicación del teorema de Stone-Weierstrass en las Redes neuronales de Chebychev	323
C.0.1. Demostración del teorema de Stone-Weierstrass para un espacio de entrada unidimensional	324

C.0.2. Demostración del teorema de Stone-Weierstrass para un espacio de entrada multidimensional	328
D. Zonas fuente en el método neuronal basado en Redes neuronales de Chebychev	335
Índice alfabético	345
Glosario	349
Bibliografía	367
Índice de figuras	377
Índice de tablas	396

Capítulo 1

Introducción

1.1. Visión general del trabajo y objetivos de la tesis

La industria de las telecomunicaciones en la actualidad demanda nuevas herramientas software de análisis de circuitos de microondas. Estas nuevas herramientas deben ser capaces, tanto de analizar con precisión dispositivos de microondas de gran complejidad como de proporcionar la respuesta deseada con rapidez. Las herramientas de análisis son necesarias para poder sintetizar o diseñar un determinado circuito de microondas. Mediante la simulación informática de la respuesta del circuito se pueden calcular las dimensiones físicas, parámetros o características del circuito que permiten alcanzar la respuesta deseada. El simulador electromagnético debe tener en cuenta los fenómenos físicos que influyen en el funcionamiento del circuito para que la respuesta obtenida en el proceso de síntesis se ajuste a la respuesta real del circuito fabricado. Si la herramienta de análisis es fiable y representa con fidelidad la respuesta real del circuito, el tiempo y gasto empleados en la tarea de diseño se puede reducir drásticamente. La fabricación de un circuito constituye un proceso mucho más caro y en ocasiones más lento que el análisis electromagnético realizado mediante herramientas informáticas. Si la respuesta del circuito fabricado no concuerda con la respuesta deseada será necesario realizar un proceso de sintonización (“tuning”). En caso de que este proceso no sea posible se puede recurrir a la fabricación de distintos prototipos hasta que alguno de ellos logre las especificaciones pretendidas. Si no se dispone de una herramienta de simulación o modelo teórico preciso, la predicción de la respuesta electromagnética será incorrecta. Así, el proceso final de fabricación será costoso y largo, bien debido a la necesidad de una etapa de sintonización, o por la necesidad de fabricación de múltiples versiones del circuito de microondas.

La precisión de la herramienta de análisis es un requisito indispensable para conseguir que el diseño de circuitos de microondas sea fiable. Asimismo es deseable que el simulador electromagnético obtenga la respuesta circuital con rapidez. Durante el proceso de síntesis se deben realizar múltiples análisis electromagnéticos para observar la evolución de la respuesta. Una de las opciones para acortar el tiempo empleado en la tarea de diseño radica en acelerar el análisis del circuito de microondas. Generalmente las técnicas de diseño más eficaces requieren de un número elevado de evaluaciones de la respuesta del circuito. Un simulador lento provocará que el proceso de síntesis sea poco práctico debido a la gran cantidad de tiempo necesaria para la obtención de una solución aceptable. Por

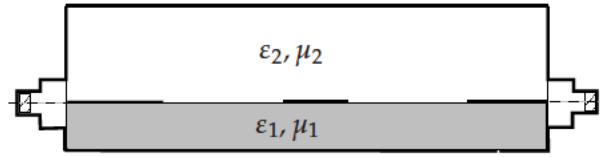


Figura 1.1: Estructura multicapa apantallada estudiada en este trabajo.

el contrario, si se dispone de una herramienta de simulación suficientemente veloz se podrán usar técnicas potentes que alcancen soluciones óptimas. Si la rápida herramienta de simulación se emplea con técnicas menos potentes que requieran pocas iteraciones para llegar a una solución, que probablemente no será óptima, el tiempo de diseño será muy pequeño. En ambos casos es apropiado utilizar una herramienta de simulación veloz y precisa.

Para que el proceso de diseño sea exitoso el circuito de microondas deberá llegar a cumplir una serie de especificaciones en el ancho de banda de trabajo. Generalmente se parte de un prototipo inicial que no cumple dichas especificaciones. En cada iteración del proceso de diseño se intentan minimizar las diferencias entre la respuesta del prototipo y las especificaciones o la respuesta del circuito ideal. El proceso finaliza cuando el circuito cumpla todas las especificaciones o se alcance el número máximo de iteraciones. De esta forma el problema de diseño suele presentarse como un problema de optimización, ya que implica la minimización de una serie de diferencias.

Antes de elegir un simulador electromagnético y un método de síntesis es indispensable elegir la tecnología adecuada para el propósito que se desea que cumpla el circuito de microondas a diseñar. Existen dos tipos principales de tecnología utilizados en la fabricación de circuitos de microondas; por un lado la tecnología basada en guías de onda y por otro la tecnología basada en estructuras microtira. La tecnología planar en general y la microtira en particular posee varias características provechosas como por ejemplo bajo coste, tamaño reducido, ligereza, fiabilidad, facilidad de integración con dispositivos y circuitos electrónicos, facilidad de fabricación y de repetición. Gracias a estas propiedades la tecnología microtira ha tenido un gran crecimiento en su utilización en áreas tales como las comunicaciones vía satélite o las comunicaciones móviles. En un principio las estructuras microtira fueron únicamente empleadas como líneas de transmisión así como filtros de microondas. Posteriormente la radiación provocada por las discontinuidades existentes en las tiras metálicas impresas sobre el dieléctrico permitió el uso de las estructuras microtira como antenas. Las antenas basadas en esta tecnología no pueden estar totalmente encapsuladas debido a la necesidad de radiar al exterior. Otros circuitos microtira tales como los filtros de microondas pueden o no estar encapsulados en una cavidad metálica. En el caso de que el encapsulamiento sea total las pérdidas por radiación de la estructura son cero. El teorema de reciprocidad indica que ninguna señal exterior podrá penetrar en la zona interior encapsulada, de forma que la estructura microtira estará totalmente apantallada. Esta protección frente a interferencias externas constituye una de las razones para encapsular los circuitos impresos en estructuras microtira. Gracias al aislamiento se podrán usar en entornos contaminados electromagnéticamente. Además, el encapsulamiento provee a la estructura de un soporte mecánico del que carecen los circuitos no encapsulados. En la Figura 1.1 podemos observar una estructura multicapa (microtira) compuesta por diferentes capas de dieléctrico con tiras metálicas impresas y encapsulada totalmente por una caja metálica.

La presente tesis tiene como objetivo el desarrollo de una herramienta software capaz de analizar con precisión y rapidez circuitos de microondas impresos en tecnología microtira y encapsulados en cajas metálicas como el mostrado en la Figura 1.1. La herramienta está basada en un conjunto de redes neuronales que se encargarán de aproximar las funciones de Green de los potenciales relevantes de la estructura multicapa apantallada. Para evaluar las funciones de Green necesarias se recurre a una formulación en el dominio espacial. Una vez entrenadas las redes neuronales necesarias para aproximar a las funciones de Green, los valores de salida de dichas redes neuronales sustituirán a los valores de las funciones de Green en la evaluación de la Ecuación integral ("Integral Equation" o IE según las siglas inglesas). Gracias a la rapidez de cálculo de los valores de salida de las redes neuronales, el análisis de los circuitos de microondas se verá claramente acelerado. El objetivo es doble, ya que se desea conseguir una serie de redes neuronales que aproximen de forma muy precisa las funciones de Green de los potenciales relevantes, y que al mismo tiempo sean muy veloces en la obtención de los valores deseados.

El segundo objetivo de esta tesis consiste en la utilización de la nueva técnica de análisis desarrollada en métodos de diseño novedosos que permitan la síntesis de mejores circuitos de microondas en tecnología microtira y encapsulados. Muchas de estas técnicas de diseño, utilizadas ampliamente en otros campos de la ciencia, no han sido aplicadas hasta ahora en el diseño de circuitos de microondas microtira, debido a la lentitud de las actuales herramientas de análisis. Este hecho es debido a que las técnicas de diseño más avanzadas necesitan simular la respuesta eléctrica del circuito un número elevado de veces. De esta forma el diseño de circuitos de microondas microtira ha quedado limitado hasta el momento únicamente a las técnicas menos eficaces. La nueva herramienta de simulación obtendrá más rápidamente la respuesta eléctrica con la precisión deseada, de forma que se podrán utilizar técnicas de diseño más potentes que las empleadas hasta la actualidad.

Generalmente el diseño de un circuito de microondas implica la optimización de un prototipo inicial cuya respuesta puede estar alejada de la respuesta ideal o deseada. El proceso de optimización finaliza cuando las especificaciones pre-determinadas han sido alcanzadas, o cuando la respuesta circuital se considera similar a la deseada. Si el mecanismo de optimización es suficientemente poderoso a la hora de encontrar soluciones óptimas, incluso a pesar de partir desde una respuesta muy distinta a la deseada, se podrá alcanzar una solución satisfactoria. Con las nuevas técnicas de optimización se podrá lograr una mejora en el diseño de circuitos de microondas, ya que se evitarán soluciones sub-óptimas que sólo alcancen a satisfacer una parte de las especificaciones. La solución óptima alcanzada por las novedosas técnicas de optimización representa por tanto un circuito de microondas con un mejor cumplimiento de las especificaciones, o con una respuesta muy cercana a la ideal. En la presente tesis se ha aplicado a la tecnología microtira apantallada la técnica de optimización conocida como Algoritmos genéticos. Esta familia de algoritmos de optimización permite alcanzar soluciones óptimas superando a las tradicionales técnicas basadas en el cálculo del gradiente. Además, se ha desarrollado una nueva función de rendimiento que posibilita una correcta operación del algoritmo de optimización. La segunda técnica de síntesis de filtros de microondas microtira y encapsulados emplea directamente redes neuronales para calcular las dimensiones del filtro. Esta técnica está basada en la aproximación, por medio de dichas redes neuronales, de las relaciones entre las dimensiones físicas de los resonadores que conforman el filtro y los coeficientes de acoplo y factores de calidad de dichos resonadores. Ambas técnicas han sido probadas principalmente en el diseño de filtros de

microondas encapsulados.

1.2. Métodos de análisis de estructuras apantalladas de microondas

El análisis electromagnético de cualquier estructura requiere la resolución de las ecuaciones de Maxwell aplicando las condiciones de contorno adecuadas. La resolución analítica de dichas ecuaciones se ve limitada a un número reducido de estructuras. Para conseguir una solución analítica las estructuras a analizar generalmente deberán estar compuestas por geometrías canónicas y en ocasiones habrá que realizar ciertas simplificaciones tales como considerar alguna dimensión de la estructura como infinita. En un principio, para poder modelizar y analizar aquellas estructuras en las que no existe solución analítica se recurrió a la identificación de ciertas partes del circuito de microondas con elementos circuitales concentrados. Así las discontinuidades en guías de onda se modelaron con elementos concentrados [Marcuvitz, 1951]. Esta técnica se amplió a circuitos basados en líneas de transmisión planares [Barret, 1955]. Más adelante se presentó un procedimiento para modelar como elementos circuitales no sólo las discontinuidades en guíaonda sino también estructuras completas basadas en guías de onda [Eisenhart y Khan, 1971]. El enfoque anterior consistía en considerar que una estructura de microondas puede soportar diversos modos electromagnéticos de forma que cada uno de ellos puede ser representado por un circuito eléctrico. Para lograr un modelo completo, el circuito eléctrico final debe incluir el acoplamiento entre los diversos modos electromagnéticos. Este procedimiento se puede aplicar a otras estructuras distintas de las guías onda. Las estructuras distribuidas también fueron modelizadas como elementos circuitales mediante el formalismo presentado en [Collin, 1966].

Los métodos anteriores resultaron de gran utilidad para analizar estructuras simples de microondas. Conforme creció la complejidad de los circuitos dichas estrategias de análisis dejaron de ser útiles debido a la falta de precisión en la obtención de la respuesta. Esta falta de precisión es debida en gran medida a que los métodos basados en elementos concentrados son solo parcialmente, y en ocasiones en absoluto, capaces de reflejar el acoplo electromagnético entre aquellos elementos circuitales cercanos en términos electromagnéticos. La solución propuesta para analizar y sintetizar nuevas estructuras de mayor complejidad consistió en resolver de forma numérica las ecuaciones de Maxwell. Una de las primeras técnicas aplicadas en la resolución numérica de las ecuaciones de Maxwell fue la de Diferencias finitas en el dominio del tiempo ("Finite difference time domain" o FDTD según las siglas inglesas) [Yee, 1966]. Este método consiste en la descomposición del sistema general de ecuaciones diferenciales en un sistema escalar de ecuaciones diferenciales y en la posterior descomposición de las derivadas parciales por derivadas aproximadas mediante diferencias finitas (perturbación en torno a un punto). Una vez obtenido el sistema de ecuaciones en diferencias finitas se pueden aplicar las condiciones de contorno del problema para finalmente proceder a su resolución numérica.

Otro de los métodos ampliamente utilizados para el modelado electromagnético es el procedimiento conocido como Elementos Finitos ("Finite elements method" o FEM según las siglas inglesas). Podemos citar [Silvester, 1969] como el primer trabajo que aplicó dicha técnica al análisis electromagnético de una estructura. La estrategia de los FEM consiste en dividir el espacio de trabajo en una serie de polígonos en el caso bidimensional o en una serie de poliedros en el caso tridimensional. En cada

polígono o poliedro la función solución se aproxima mediante una serie de polinomios u otro tipo de funciones base multiplicados por coeficientes desconocidos. El problema por tanto radica en calcular los coeficientes que permiten la correcta aproximación de la solución en una serie de puntos discretos situados en los polígonos o poliedros que dividen el espacio. La principal ventaja de los métodos anteriores radica en que permiten estudiar multitud de estructuras tanto si están encapsuladas o no. Son métodos muy versátiles ya que pueden analizar estructuras complejas de sección arbitraria con una, varias o ninguna capa de dieléctrico. Su mayor desventaja estriba en que hay que discretizar todo el espacio de trabajo de modo que el número de incógnitas puede llegar a ser muy elevado. Por ejemplo, en el caso de la técnica FEM será necesaria la utilización de grandes matrices aunque este problema es cada vez menos importante debido a que los ordenadores poseen memorias de gran tamaño y de pequeño coste. Las matrices que se obtienen en el método FEM suelen ser dispersas, no densas, de forma que se pueden aplicar algoritmos eficientes para su resolución. A pesar de este hecho los tiempos de computación en el método FEM son elevados, de forma que su utilización en herramientas de diseño puede llegar a ser problemática.

Una de las técnicas más exitosas en el análisis de circuitos de microondas es la conocida como Ecuación integral (IE). En este método los campos electromagnéticos se representan mediante las funciones de Green del medio de trabajo. Las funciones de Green constituyen en el análisis electromagnético la respuesta del sistema a la excitación impulso. La mencionada respuesta del sistema viene determinada por las ecuaciones de Maxwell y la excitación impulso corresponde a una corriente unitaria puntual de tipo eléctrico o magnético. La utilización de las funciones de Green, junto con la imposición de las condiciones de contorno adecuadas, permite la obtención de una Ecuación integral, o sistema de ecuaciones integrales, que al constituir un operador lineal se puede resolver gracias a la aplicación del Método de los momentos ("Method of Moments" o MoM según las siglas inglesas) [Harrington, 1987, Harrington, 1993]. En el MoM las corrientes inducidas desconocidas son aproximadas por un sumatorio de funciones conocidas multiplicadas cada una de ellas por un coeficiente. Los coeficientes se convierten así en las incógnitas, que una vez despejadas permiten conocer con un cierto grado de precisión las corrientes deseadas. Las corrientes incógnita solo existen en una parte del espacio de trabajo, por ejemplo, en las líneas metálicas en un circuito impreso en tecnología microtira. Para simplificar el cálculo de dichas corrientes, el espacio en el que se encuentran es usualmente dividido en una serie de sub-secciones. De este modo la aproximación de cada corriente en cada sub-sección es llevada a cabo por una función de base que existe únicamente en una sub-sección determinada. A estas funciones de base se les denomina de dominio parcial. Por tanto, se produce una división en polígonos o mallado del espacio al igual que en el método de Elementos Finitos, pero en este caso únicamente se debe operar sobre aquella región del espacio que contenga corrientes eléctricas o magnéticas. Así, el tamaño de las matrices resultantes es mucho menor que en el método FEM por lo que, a pesar de que dichas matrices son densas, el tiempo de computación destinado a resolver los sistemas será a su vez mucho menor.

En el método de la IE (ver [Mosig, 1989]) la mayor cantidad de tiempo se destina a evaluar las funciones de Green del medio en cuestión para medios complejos. Para reducir el tiempo empleado en la IE se suele actuar sobre el cálculo de las funciones de Green. Una reducción en el tiempo invertido en el cálculo de las funciones de Green apropiadas redundará en una importante disminución del tiempo requerido para completar los elementos de las matrices involucradas en el MoM.

Por tanto, si las funciones de Green pueden ser calculadas en un determinado medio de forma eficiente, el análisis mediante la IE requerirá menos recursos informáticos y será más veloz en general que el realizado tanto con FDTD como con FEM. Por ello se ha llegado a combinar las técnicas anteriores con la IE, para aligerar la carga computacional que conlleva el análisis de dispositivos como las antenas de microondas. Así, en el método conocido como “Hybrid finite element method” (HFEM según las siglas inglesas) una parte del espacio se analiza mediante el FEM, mientras que la región correspondiente al espacio exterior a la anterior región es analizada con la IE [Yuan et al., 1990]. De esta forma la discretización llevada a cabo por el FEM se limita a una región reducida del espacio, por lo que el sistema de ecuaciones que se debe resolver se ve asimismo reducido. Una desventaja de la IE respecto a las técnicas de FDTD y FEM radica en la limitación de la IE a aquellas estructuras cuyas funciones de Green de los campos o de los potenciales electromagnéticos son conocidas. Sin embargo, la utilización de la función de Green de espacio libre ha permitido el análisis de estructuras muy diversas y de un cierto grado de complejidad.

En lo que respecta al análisis de estructuras apantalladas multicapa mediante la IE, la influencia de las diversas capas de dieléctrico es proporcionada por las funciones de Green. Existen dos formulaciones de las funciones de Green del medio apantallado, la formulación en el dominio espacial y la formulación en el dominio espectral. La determinación de las funciones de Green en el dominio espectral es simple, ya que implica el cálculo de un conjunto de coeficientes que son analíticos. En cambio, la evaluación de las funciones de Green en el dominio espacial involucra el cálculo de series infinitas de modos que se propagan en el medio. Dichas series dependen de funciones sinusoidales que oscilan ampliamente. Por ello, la evaluación de dichas series infinitas convergerá lentamente. Cada formulación conlleva la definición del sistema de ecuaciones integrales en el dominio correspondiente. En el dominio espacial, en primer lugar, se evalúan las funciones de Green, es decir, sus series asociadas. Una vez evaluadas se utilizan en la resolución de la IE. En cambio, en el dominio espectral, las series asociadas se introducen en la formulación de la IE. De este modo, los elementos de la matriz de impedancias generalizadas van a depender de las series espectrales. Este hecho produce una de las principales desventajas de la formulación de la IE en el dominio espectral frente a la formulación en el dominio espacial. Si la cavidad es eléctricamente más grande que las metalizaciones impresas, entonces serán necesarios muchos modos para que converjan las series que conforman los elementos de la matriz de impedancias generalizadas. Por el contrario, la formulación espacial no depende del tamaño relativo de la cavidad respecto a las metalizaciones, ya que las funciones de Green son utilizadas en la IE una vez han sido calculadas. La formulación de la IE en este dominio es mucho más sencilla que en el dominio espectral.

Por las anteriores razones, la reducción del tiempo empleado en el cálculo de las funciones de Green implica un gran ahorro en el tiempo de evaluación de la IE. Durante las últimas décadas se han desarrollado numerosos trabajos encaminados primero a posibilitar, y posteriormente a acelerar el cálculo de las funciones de Green en diversas estructuras tanto apantalladas como no apantalladas así como multicapa y no multicapa. Entre los primeros trabajos dedicados al análisis de estructuras apantalladas mediante la IE podemos citar [Rautio y Harrington, 1987], [Dunleavy y Katehi, 1988a, Dunleavy y Katehi, 1988b] y [Railton y Meade, 1992]. Una de las primeras técnicas empleadas para acelerar el cálculo de las funciones de Green del medio encapsulado multicapa consistió en aplicar el algoritmo de la transformada rápida de Fourier (“Fast Fourier transform” o FFT según las si-

glas inglesas) en la evaluación de las suma modales que componen los elementos de la matriz de impedancias surgidos del MoM en la IE en el dominio espectral [Rautio, 1986, Hill y Tripathi, 1991]. Para aplicar la FFT es necesario que las series infinitas sean truncadas, de modo que aumenta el riesgo de “aliasing”. La FFT puede ser combinada con otras técnicas numéricas para aumentar su eficiencia. Así, en [Hill y Tripathi, 1991] se computan únicamente cierta cantidad de elementos de la matriz de momentos calculando los restantes mediante una interpolación lineal de los ya calculados. De este modo, las transformadas de Fourier correspondientes a los elementos de la matriz interpolados no se realizan, reduciendo el número total de transformaciones.

Una de las mayores limitaciones de la FFT radica en la necesidad de dividir el circuito en subsecciones uniformes. Para conseguir un análisis preciso, la discretización debe ser densa en ciertas zonas del circuito, como por ejemplo en las discontinuidades. Si el mallado debe ser uniforme en todo el circuito, entonces la discretización deberá ser igualmente densa tanto en zonas en las que no sea realmente necesario una mayor subdivisión, como en aquellas en las que una discretización más densa sí esté justificada. Por tanto un mallado uniforme suele ser menos eficiente que uno no uniforme. Este inconveniente puede sortearse mediante el uso de la transformada de Fourier no-uniforme (“Non uniform fast Fourier transform” o NUFFT según las siglas inglesas), que permite muestrear en puntos situados de forma arbitraria tanto en el dominio espacial como en el dominio transformado. En [Su y Kuo, 2005] la NUFFT bidimensional es empleada para analizar diversos circuitos de tipo microtira encapsulados. Los valores de la NUFFT bidimensional son obtenidos a partir de la interpolación de los valores de una FFT uniforme sobre-muestreada en la misma región de interés. El sobre-muestreo de la FFT uniforme debe ser suficiente para que la aproximación de los valores de la NUFFT alcance cierta precisión. Aunque la NUFFT, al igual que la FFT, impone que cada subdivisión sea rectangular, se consiguen mallados muy flexibles y mejor adaptados a la geometría del circuito que los obtenidos con la FFT.

A pesar de la flexibilidad de la NUFFT, la discretización de circuitos complejos que incorporen tramos circulares puede llegar a ser muy difícil cuando no imposible. Para alcanzar una mayor libertad en la discretización del circuito, y al mismo tiempo conseguir una reducción en el tiempo de análisis, hay que actuar sobre la evaluación de las funciones de Green sin recurrir al empleo de la FFT. Una de las técnicas más empleadas consiste en convertir las funciones sinusoidales que aparecen en la definición de las funciones de Green del dominio espacial, en otro tipo de funciones que no presenten el comportamiento oscilante de las primeras. Las funciones sinusoidales pueden ser transformadas en funciones de Bessel modificadas que presentan un rápido decaimiento en sus valores [Hashemi-Yeganeh, 1995]. La conversión de las funciones sinusoidales en funciones de Bessel puede ser combinada con otras técnicas para incrementar la eficiencia del cálculo de las funciones de Green. De este modo, en [Eleftheriades et al., 1996] la técnica de la extracción asintótica es utilizada para dividir las funciones de Green del dominio espacial de los potenciales del medio multicapa encapsulado en dos partes. Por un lado, las series correspondientes a la parte dinámica son evaluadas mediante el método de aceleración de series de Kummer. Por otro lado las series de la parte estática, independientes de la frecuencia, son tratadas con la fórmula de Poisson y la identidad de Sommerfeld. Gracias a la aplicación de las técnicas anteriores, las funciones sinusoidales de la parte estática, dependientes de dos dimensiones, son convertidas en funciones de Bessel modificadas de una sola dimensión que decaen rápidamente, de forma que la suma final requiere de pocos términos para su

evaluación. Asimismo, la técnica de la extracción asintótica puede aplicarse en el dominio espectral tal como se aprecia en el análisis de antenas de parche emplazadas en cavidades, desarrollado en [Gentili et al., 1997]. En este trabajo, la parte estática resultante de la extracción asintótica no es transformada en cierto tipo de funciones, sino que es convertida en una suma de imágenes espaciales que converge rápidamente. Una aplicación de la fórmula de Poisson para la transformación de series que involucran funciones sinusoidales en series de imágenes en el dominio espacial se encuentra en [Park y Nam, 1998]. Así, las series originales son convertidas en series que dependen de las funciones de Green periódicas de espacio libre. Para poder evaluar las series transformadas se emplea la técnica de Ewald [Park y Nam, 1997], que permite la transformación de las series de exponenciales complejas oscilantes en nuevas series cuyos términos dependen de funciones de error complementario cuyo decaimiento es acentuado.

En la presente tesis las funciones de Green son evaluadas, en el dominio espacial, mediante las técnicas descritas en [Melcon y Mosig, 2000]. En ese trabajo la evaluación de las funciones de Green del medio multicapa apantallado es conseguida de forma eficiente gracias a la aplicación de dos estrategias en regiones diferentes del espacio. En la zona cercana a la fuente se emplea la fórmula de Poisson para obtener un sumatorio de imágenes espaciales. De esta forma la singularidad de las funciones de Green queda representada analíticamente mediante el término de la fuente, y unas pocas imágenes espaciales. Lejos de la fuente, donde la singularidad no es importante, se aplica un procedimiento de suma por partes directamente sobre la serie sinusoidal para acelerar el cálculo en esta parte del espacio (ver [Álvarez Melcón, 1998]).

1.3. Solución propuesta para acelerar el análisis de circuitos de microondas encapsulados e impresos

En el apartado anterior se ha mostrado que para conseguir acelerar el análisis de un circuito de microondas mediante la IE es necesario reducir el tiempo de evaluación de las funciones de Green del medio. En el caso de un medio multicapa apantallado las funciones de Green en el dominio espacial se representan por medio de series que involucran funciones sinusoidales, por tanto de lento decaimiento. En el anterior apartado se han citado diversos intentos de acelerar la evaluación de las funciones de Green del medio apantallado. La solución propuesta en la presente tesis no radica en la aceleración de las funciones de Green, sino en el empleo de redes neuronales para aproximar las propias funciones de Green del medio encapsulado y multicapa. Así, las redes neuronales sustituirán a los valores de las funciones de Green evaluados numéricamente, durante la resolución de la IE. Esta es la primera vez que se propone el uso de redes neuronales para aproximar las funciones de Green dentro de un problema de Ecuación integral. A continuación vamos a describir las características que definen a las redes neuronales y la metodología empleada para acelerar el análisis de los circuitos del medio apantallado.

1.3.1. Redes neuronales

Podemos definir una red neuronal como un procesador ampliamente distribuido de forma paralela, y compuesto por unidades simples de procesado. Una red neuronal tiene una propensión natu-

ral para guardar el conocimiento y para hacer disponible su uso [Haykin, 1999]. El adjetivo neuronal hace referencia a su parecido con un cerebro humano en dos aspectos:

- La red adquiere el conocimiento de su entorno a través de un proceso de aprendizaje.
- El peso de cada una de las conexiones interneuronales, conocidas como pesos sinápticos, es empleado para almacenar el conocimiento adquirido.

El proceso de aprendizaje consiste en un algoritmo también denominado algoritmo de entrenamiento. Su función es la de modificar los parámetros libres, tales como los pesos sinápticos o los parámetros que determinan la función matemática insertada en cada neurona. Las redes neuronales pueden cambiar su topología; este aspecto es similar al nacimiento y muerte de las neuronas del cerebro. Por supuesto, el cerebro humano es mucho más complejo que cualquier red neuronal, y por tanto los puntos en común son ciertos a un nivel muy simple.

El proceso de aprendizaje de las redes neuronales es un concepto que podría ser aplicado a otras técnicas, como por ejemplo la interpolación polinómica o la teoría de construcción de filtros lineales adaptativos. El cálculo de ciertos parámetros libres en las técnicas anteriores parece en un primer momento asimilable al proceso de aprendizaje de una red neuronal. Sin embargo, el algoritmo de entrenamiento de una red neuronal posee la característica propia de generar nuevas neuronas hasta que se alcanza cierto nivel de conocimiento. Por tanto, durante el entrenamiento se añadirán nuevas neuronas mediante la evaluación de sus parámetros libres, examinando la calidad del aprendizaje cada cierto tiempo o cada cierta cantidad de neuronas.

Cada tipo de entrenamiento está fundamentado en una teoría científica o matemática diferente. Por ello, el mecanismo de generación de nuevas neuronas o entrenamiento es una de las características que diferencian las distintas clases de redes neuronales. Así, por ejemplo, la red neuronal conocida como máquina de Boltzmann se basa en la mecánica estadística, mientras que la red de Hopfield está relacionada con el estudio de sistemas dinámicos [Haykin, 1999]. Aunque otras técnicas de resolución de problemas de clasificación, procesamiento de señal o control sean susceptibles de ser estructuradas como una red neuronal, es imprescindible que el algoritmo de aprendizaje permita la adición o eliminación de módulos o neuronas cuando se considere necesario.

Todos los tipos de redes neuronales deben estar compuestos por módulos independientes llamados neuronas. Por lo tanto, la estructura de todas las redes neuronales va a ser similar. La diferenciación entre los diferentes tipos de redes neuronales está determinada bien por el tipo de entrenamiento, bien por la clase de función matemática que define la neurona, o bien por ambos. Así, redes neuronales que posean una clase distinta de neuronas, pero compartan una misma estructura, pueden compartir un mismo algoritmo de entrenamiento. Asimismo, una misma estructura con el mismo tipo de neuronas puede poseer diferentes tipos de entrenamiento, dando lugar a dos clases de redes neuronales diferentes. Por tanto, para que una técnica de adquisición de conocimiento de un entorno o problema sea catalogada como red neuronal, debe poder ser estructurada en módulos independientes y contar con un método de entrenamiento propio y adecuado.

Las redes neuronales ofrecen una serie de beneficios frente a otras técnicas a la hora de resolver problemas de clasificación o procesamiento de señal:

- Generalización: las redes neuronales son capaces de producir salidas aceptables ante la presencia de entradas nunca vistas antes. Es decir, generan datos de salida razonables a partir de datos de entrada nuevos.
- Descomposición: para que la generalización sea adecuada, habitualmente se recurre a la descomposición del problema a tratar en tareas relativamente simples. Dicha descomposición se debe establecer en un marco consistente con el problema a solucionar. En este modo de funcionamiento las redes neuronales se parecen de nuevo al modo de operación en el cerebro humano, ya que éste posee zonas especializadas para cada una de las tareas del razonamiento y la percepción. Además, también en el cerebro cada tarea se descompone en diferentes sub-tareas.
- No-linealidad: generalmente la mayor parte de los problemas que deben resolver las redes neuronales son no-lineales. Los mecanismos físicos que se desean modelar son en muchas ocasiones no-lineales. Así, la red neuronal debe presentar a su vez un carácter no-lineal. Dicha propiedad se consigue mediante el uso de neuronas no-lineales.
- Aprendizaje supervisado: este tipo de aprendizaje consiste en aprender a partir de ejemplos en los que se provee una entrada y su salida correspondiente. Los parámetros de la red neuronal se ajustan para que la red neuronal represente con cierto grado de fidelidad la relación entre los datos de entrada y los datos de salida. De este modo, se realiza un mapeo ("mapping") entre el espacio de entrada y el espacio de salida. En ocasiones solo ciertos parámetros son calculados teniendo en cuenta los valores de los datos de salida. En este caso, decimos que los parámetros calculados únicamente con los datos de entrada han utilizado un mecanismo de aprendizaje no supervisado. Los datos de entrada-salida pueden ser presentados a la red neuronal varias veces hasta que se considere que ha finalizado el entrenamiento.
- Adaptabilidad: las redes neuronales poseen la capacidad de cambiar sus parámetros libres, como por ejemplo sus pesos, para adaptarse a los cambios que surjan en su entorno. El cambio en los parámetros estriba generalmente en un re-entrenamiento en el que se emplean los nuevos datos correspondientes al entorno cambiante. La estructura de ciertos tipos de redes neuronales permite con mayor facilidad la operación de éstas en entornos no-estacionarios. Sin embargo, la capacidad de adaptación no solo es provista por la estructura de la red neuronal sino también por el algoritmo de entrenamiento. En general, cuanto más adaptativo sea el funcionamiento de la red neuronal, mejor será su rendimiento en entornos no estacionarios, siempre y cuando conserve la estabilidad.
- Estructura paralela: toda red neuronal posee una arquitectura en forma de módulos conectados en paralelo. De esta forma los cálculos realizados en cada una de las neuronas son independientes del resto de neuronas. Esta característica provee a las redes neuronales de dos propiedades muy deseables. En primer lugar presentan una alta tolerancia a fallos. Si aparece un fallo en una neurona o en alguna de sus conexiones, la respuesta se degradará levemente. Si las neuronas estuviesen dispuestas en una estructura serie, la tolerancia a fallos sería muy baja, debido a la propagación del error por todas las neuronas. Una alteración o fallo en una neurona puede

afectar a aquellas neuronas conectadas a ella. Sin embargo, la respuesta, al depender de todo el conjunto de neuronas, no sufrirá un fallo catastrófico. En segundo lugar, gracias no solo a la estructura paralela, sino también a la tolerancia a fallos, las redes neuronales son muy aptas para su implantación en tecnología de tipo VLSI (“very large-scale integrated technology”). Además, la paralelización de los cálculos en la red neuronal permite que la respuesta sea rápidamente calculada.

- Uniformidad en el análisis y diseño: a pesar de la heterogeneidad de las teorías empleadas para generar los diferentes tipos de redes neuronales, todas ellas comparten una estructura similar en forma de neuronas inter-conectadas. Esta propiedad permite que diferentes clases de algoritmos de entrenamiento sean en ocasiones compartidas por distintas redes neuronales. La uniformidad posibilita además la construcción de redes modulares a través de la integración directa de módulos.

1.3.2. Clases de redes neuronales: visión histórica

El artículo pionero en el campo de la cibernética y las redes neuronales data de 1943. En [McCulloch y Pitts, 1943] se describió por primera vez una red neuronal formada por neuronas simples inter-conectadas y operando de forma sincronizada. Se demostró que una red constituida de esta forma podía resolver en principio cualquier función computable.

Hasta la década de los 50 no aparece el primer intento de realizar una simulación de una red neuronal en un ordenador. Así en [Rochester et al., 1956] se implantó en un ordenador la teoría de aprendizaje de Hebb [Hebb, 1949]. Esta teoría, siguiendo una idea de Ramón y Cajal, postulaba que la efectividad de una sinapsis se refuerza cada vez que alguna de las neuronas a las que está conectada se activa. Ese mismo año se diseñó una red neuronal con sinapsis o pesos modificables, capaz de clasificar conjuntos simples de patrones binarios [Uttley, 1956]. Los avances continuaron hasta el final de la década. Es muy destacable la introducción por parte de Rosenblatt de un nuevo método de aprendizaje supervisado basado en una unidad de cálculo a la que denominó perceptrón [Rosenblatt, 1956]. En [Rosenblatt, 1960] se demostró el teorema de convergencia del perceptrón. Esta red neuronal se convirtió, debido a estos avances, en la más exitosa de la década de los 60, dando lugar a multitud de aplicaciones. Esta década fue fructífera para el campo de las redes neuronales, ya que surgieron además otros tipos de redes como Adaline (“Linear Adaptative Element” o elemento lineal adaptativo) [Widrow y Hoff, 1960]. Una evolución de la anterior red neuronal se convirtió en la primera red neuronal formada por más de una capa de neuronas. En efecto, Madaline (“Multiple Adaline”) estaba formada por varias capas con múltiples elementos adaptativos [Widrow, 1962].

Sin embargo, el desarrollo de nuevas redes neuronales y de sus aplicaciones se frenó en la década de los años 70 debido a diversos factores. Por un lado se incumplieron las elevadas expectativas creadas durante la década anterior. Las redes neuronales podían ser aplicadas en multitud de tareas, pero estaban muy lejos de alcanzar las capacidades del cerebro humano. Por otro lado, en [Minsky y Papert, 1956] se demostró que una red neuronal basada en una única capa de perceptrones poseía límites fundamentales en su capacidad de cálculo. Además, los autores afirmaron que no existían razones para suponer que un Perceptrón multicapa (“Multilayer Perceptron” o MLP según las siglas inglesas) podría superar las limitaciones de un perceptrón de una sola capa. Finalmente,

existían pocos ordenadores disponibles para realizar las simulaciones necesarias para avanzar en el desarrollo de nuevas redes neuronales. Por tanto, solo unos pocos investigadores continuaron el estudio de las redes neuronales. Cabe destacar los trabajos sobre mapas auto-organizativos desarrollados en [der Malsburg, 1973] y [Willshaw y der Malsburg, 1976]. Los mapas auto-organizativos están inspirados por la forma en la que se organizan topológicamente las diferentes áreas cognitivas del cerebro.

En la siguiente década, gracias a las mejoras tecnológicas, los estudios relacionados con redes neuronales tomaron una nueva importancia. Así, se retomaron los estudios relacionados sobre mapas auto-organizativos surgidos en los años 70. En 1980 apareció la red neuronal basada en la teoría de resonancia adaptativa ("Adaptive Resonance Theory" o ART según las siglas inglesas) [Grossberg, 1980]. Esta red utiliza una capa de reconocimiento y una capa superior de generación de salida. En [Kohonen, 1982] se propuso un nuevo desarrollo de mapa auto-organizativo que se ha convertido en la referencia en este tipo de redes neuronales. Dicho mapa está compuesto por una rejilla unidimensional o bidimensional de neuronas que se ajustan mediante un modelo de aprendizaje no supervisado. A principios de la década se introdujo un nuevo enfoque en el análisis de redes neuronales recurrentes, gracias al uso de una función de energía para definir su funcionamiento y de la aplicación de teorías provenientes de la física [Hopfield, 1982]. Las redes recurrentes son aquellas en las que aparece una realimentación desde la salida de datos hasta la entrada. A partir del trabajo mencionado anteriormente, las redes recurrentes suelen ser denominadas como redes de Hopfield. En 1985 el empleo de la teoría de optimización "Simulated Annealing" sirvió para el desarrollo de la primera red multicapa que realmente tuvo éxito en la solución de problemas [Ackley et al., 1985]. A esta red se le denominó máquina de Boltzmann y demostró que la hipótesis de Minsky y Papert no era cierta. Un año más tarde se desarrolló uno de los métodos de aprendizaje más empleados, el algoritmo de retro-propagación ("back-propagation algorithm") [Rumelhart et al., 1986]. Dicho método ha sido usado hasta el momento presente en multitud de aplicaciones sobre todo en perceptrones multicapa. Una alternativa a esta red neuronal surgió en 1988 con las Redes neuronales basadas en funciones de base radial ("Radial Basis Function Neural Networks" o RBFNN según las siglas inglesas) [Broomhead y Lowe, 1988]. Dicha técnica tiene sus raíces en el método de funciones potenciales. Además, en 1990 Poggio y Girosi aplicaron a las RBFNNs la teoría de regularización de Tikhonov, mejorando su rendimiento en ciertas situaciones, como por ejemplo en presencia de ruido [Poggio y Girosi, 1990]. En la presente tesis se emplean este tipo de redes para aproximar las funciones de Green del medio multicapa. En el Capítulo 3 se explican las razones para emplear este tipo de redes en lugar de otras descritas en este apartado.

A principios de la década de los 90 aparecieron las Máquinas de vectores soporte ("Support Vector Machines" o SVM según las siglas inglesas) cuyo funcionamiento está basado en la teoría del aprendizaje con conjuntos de tamaño finito [Vapnik, 1992, Cortes y Vapnik, 1995]. El diseño de una SVM está relacionado con la dimensión de Vapnik-Chervonenkis (VC), que provee una medida de la capacidad de una red neuronal de aprender a partir de un conjunto de ejemplos. Durante las dos últimas décadas se han producido numerosos avances en la mejora del proceso de entrenamiento de diferentes tipos de redes neuronales. Además, el número de aplicaciones en las que este conjunto de técnicas ha mostrado su validez es elevado. El campo de aplicación de las redes neuronales es muy amplio, abarcando disciplinas muy variadas.

En el presente trabajo no solo se ha empleado una red neuronal conocida como RBFNN, sino que también se ha desarrollado una nueva red neuronal cuyas prestaciones mejoran los resultados obtenidos con las RBFNNs. Esta nueva red emplea polinomios de Chebychev y es similar a las RBFNNs, tanto en su estructura como en su método de entrenamiento. Hasta donde llega el conocimiento del autor, es la primera vez que se emplean estos polinomios en una red neuronal con el método de entrenamiento que será descrito en el Capítulo 4.

1.3.3. Metodología

Tal y como se ha explicado en los dos apartados anteriores, las redes neuronales son capaces de aprender de su entorno. Este aprendizaje puede asemejarse a la captura de las relaciones entre una serie de datos de entrada y unos datos de salida. Los datos de entrada y salida pueden pertenecer a distintos espacios multidimensionales, de modo que distinguiremos entre espacio de entrada y espacio de salida. Debido al comportamiento no lineal de las neuronas, las redes neuronales podrán realizar un mapeo entre espacios de entrada y salida de forma no lineal. Estas habilidades son utilizadas en dos tipos de problemas diferentes. Por un lado existen problemas de clasificación, en los que las salidas pertenecen a una serie de valores discretos. Por otro lado, en los problemas de regresión las salidas pertenecen a un subconjunto de los números reales.

Los problemas de regresión se asimilan a la representación o aproximación de una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ que puede ser o no conocida. Si la función es desconocida el interés estriba en que la red neuronal modele un proceso físico del que se poseen una serie de datos. En cambio, si la función es conocida el interés en el uso de una red neuronal se centra generalmente en la sustitución de la función conocida por parte del modelo neuronal. En este caso, la red neuronal constituye un modelo sencillo de la función conocida. El cómputo de la salida del modelo neuronal será fácil, poco costoso y muy rápido. Cuando se presentan nuevos datos de entrada, la red neuronal deberá devolver valores de salida adecuados, es decir, deberá generalizar correctamente. Por tanto, ante un nuevo valor de entrada el modelo neuronal realiza una interpolación entre los valores de entrada-salida ya conocidos para calcular el valor de salida apropiado. De esta manera, las redes neuronales se definen como interpoladores multidimensionales no lineales. En la presente tesis, las redes neuronales se emplean por primera vez para aproximar una serie de funciones conocidas, las funciones de Green del medio. Los modelos neuronales generados proveerán los valores necesarios para resolver la IE de manera mucho más veloz, que con el método de evaluación numérica directa de las funciones de Green.

Las redes neuronales deben ser entrenadas para que capturen las relaciones entre el espacio de entrada y salida. Como se explicó en la Sub-sección 1.3.1, el entrenamiento consiste en el cálculo de los parámetros libres de la red neuronal. En primer lugar será necesario generar los pares de datos de entrada-salida para hallar los parámetros libres. Generalmente los datos de entrada se encuentran acotados en el interior de cierta región del espacio correspondiente. Una vez delimitada la región de interés puede dar comienzo el proceso de generación de datos. Para conseguir una correcta generalización el conjunto de datos se divide en dos grupos diferentes. Un grupo constituirá el conjunto de datos de entrenamiento. El segundo grupo, distinto al anterior, es el conjunto de datos de validación o test. Durante el entrenamiento los parámetros libres se calcularán con los datos correspondientes, y la generalización se comprobará con los datos de validación. El entrenamiento finaliza cuando se

alcanza un nivel de error determinado. Además, con el objetivo de mejorar la calidad de la aproximación, el espacio de entrada se dividirá en distintas sub-regiones, de forma que cada modelo neuronal estará especializado en una única región. De una forma simple, se muestra a continuación la metodología empleada:

- Determinación de las variables de entrada y salida. Las variables de salida siempre son las partes reales e imaginarias de las funciones de Green. Las variables de entrada están determinadas por las características del medio encapsulado multicapa. Por ejemplo, la frecuencia puede ser o no una variable de entrada dependiendo de si se desea aproximar las funciones de Green en un ancho de banda de trabajo, o para una única frecuencia, respectivamente.
- Acotación de la región de interés del espacio de entrada. Está determinado por las características del medio encapsulado. Por ejemplo, si una caja metálica posee unas dimensiones de $a \times b$ mm la variable x correspondiente a una abscisa podrá extenderse desde 0 mm hasta a mm. En cambio la variable y se extenderá desde 0 mm hasta un máximo de b mm.
- Proceso de división en sub-regiones del espacio de entrada. Este es un proceso optativo, aunque generalmente inevitable, que tiene por objetivo la mejora del entrenamiento. El proceso de división genera de forma automática zonas con una densidad de puntos de entrenamiento adecuada para alcanzar la precisión deseada. Además, se han tomado en cuenta las características del problema de aproximación para diseñar zonas apropiadas en cada una de las coordenadas del espacio de entrada.
- Entrenamiento de las redes neuronales. La tarea de la aproximación de las funciones de Green es costosa, por tanto será imprescindible la generación de varias redes neuronales. El entrenamiento incluye una fase de cálculo de los parámetros libres, y otra fase de comprobación de la generalización de la red neuronal. Para conseguir un entrenamiento preciso es imprescindible eliminar la singularidad de la fuente puntual. Para ello, en la presente tesis, se han aplicado dos métodos diferentes. El primero consiste en la multiplicación de las funciones de Green por la distancia (ρ) entre punto fuente y punto observación. El segundo procedimiento, es mucho más efectivo, y está basado en la extracción de parte de las imágenes espaciales que aparecen en la formulación espacial.
- Implantación de las redes neuronales entrenadas en el software que evalúa la IE mediante el MoM.

1.4. Nuevos métodos de diseño de circuitos de microondas encapsulados y apantallados.

Los nuevos métodos de diseño desarrollados en la tesis hacen uso de la nueva herramienta de análisis basada en redes neuronales. Debido a la aceleración de la simulación de estructuras impresas en un medio multicapa apantallado, se podrán aplicar nuevas técnicas de diseño. Muchas de las técnicas tradicionales de diseño de circuitos de microondas emplean una serie de ecuaciones basadas en datos empíricos para calcular un primer prototipo del circuito. Las ecuaciones empíricas proveen

el circuito inicial con rapidez, pero debido al avance en la integración de circuitos son incapaces de alcanzar la precisión deseada [Hong y Lancaster, 2001]. Otra solución para encontrar un prototipo inicial radica en el empleo de curvas que relacionan los parámetros físicos buscados con los parámetros eléctricos que se desean sintetizar [Hong y Lancaster, 2001]. Las curvas permiten calcular a un ingeniero las dimensiones de un circuito con facilidad, siempre que relacionen pocas variables. Poseen la desventaja de necesitar un número relativamente alto de datos para llegar a cierta precisión. Además, impiden la automatización del proceso de diseño, al requerir la presencia de un experto que halle los valores de la estructura.

Generalmente, la etapa final del diseño implica la optimización del circuito inicial a fin de conseguir el cumplimiento de todas las especificaciones deseadas [Bandler et al., 1985, Bandler et al., 1993]. El proceso de optimización puede ser aplicado un diferentes pasos. En [Guglielmi, 1994] se desarrolló una técnica en la que el diseño del filtro final se consigue gracias a una serie de sucesivas optimizaciones de partes simples del circuito. En [Alos y Guglielmi, 1997], se emplean algoritmos basados en el gradiente de una función objetivo para optimizar paso a paso la estructura de microondas. En ambos trabajos se diseñaron filtros de microondas en guíaonda, pero la técnica puede ser aplicada en otros tipos de tecnologías. Tanto si la optimización se emplea en la etapa final como si se aplica paso a paso, generalmente se utiliza una técnica basada en el uso del gradiente de una función objetivo o función de error. La función objetivo suele depender de la diferencia entre la función deseada y la función simulada. Este tipo de procedimientos suelen emplear simuladores electromagnéticos. Estas herramientas informáticas permiten una alta precisión en comparación con las técnicas empíricas o semi-analíticas. La convergencia en el proceso de optimización hacia una solución es rápida, siempre que el circuito de partida sea apropiado. De esta forma el tiempo total de diseño puede ser reducido, aun cuando el simulador sea lento en el cálculo de la respuesta. La principal desventaja de este tipo de técnicas radica en la dependencia de la convergencia con el punto inicial elegido para realizar la optimización. Si el punto inicial no es seleccionado de forma adecuada, se puede llegar a una solución sub-óptima, que no cumpla todas o parte importante de las especificaciones.

En la década pasada se desarrolló un método que intenta aprovechar las ventajas de las técnicas empíricas y de los simuladores más precisos para diseñar circuitos de microondas. El "Space-Mapping", introducido por primera vez en [Bandler et al., 1994b], utiliza para optimizar circuitos de microondas un método de análisis poco preciso conjuntamente con un método preciso. La técnica se basa en hallar una transformación sobre los parámetros del método impreciso para calcular unos nuevos parámetros de este mismo método. La transformación debe asegurar que los nuevos parámetros, una vez introducidos en el método impreciso, ofrecen una respuesta electromagnética similar a la obtenida con el simulador preciso. De esta forma podemos emplear el método originalmente impreciso para optimizar el prototipo inicial. La clave del método reside en que el cálculo de la transformación entre los dos espacios sea fiable.

En el presente trabajo se intenta superar los problemas de las técnicas de diseño tradicionales. Para ello se han desarrollado dos técnicas con aspectos novedosos, que permiten el diseño de filtros de microondas encapsulados de forma eficaz. El primer método hace uso de la técnica de optimización de los Algoritmos genéticos ("Genetic Algorithms" o GA según las siglas inglesas, AG según las españolas). Dicha técnica permite la obtención de una solución óptima o cuasi-óptima sin

depender del prototipo inicial escogido. El segundo método posibilita la obtención de un prototipo inicial que cumple prácticamente todas las especificaciones. De esta forma, se puede considerar listo para su fabricación, o como punto inicial para un proceso fiable y rápido de optimización basada en gradiente. A continuación se explican brevemente las características principales de ambos métodos.

1.4.1. Técnica de diseño basada en Algoritmos genéticos.

Los Algoritmos genéticos pertenecen a un grupo de métodos de optimización que podemos catalogar como técnicas de optimización global. Por contra, las técnicas tradicionales basadas en el uso del gradiente se pueden catalogar como técnicas locales. El calificativo local o global está relacionado con la región del espacio de solución en el que el algoritmo busca el máximo. Entre las técnicas locales que emplean el gradiente, podemos citar los métodos de gradiente conjugado y los basados en una iteración de tipo Quasi-Newton. Existen técnicas locales que no utilizan el cálculo de ninguna derivada, como el método simplex de optimización de problemas lineales. Entre los métodos globales encontramos la técnica de “Random Walk”, la de “Simulated Annealing”, y un grupo cuyos fundamentos se hallan en la naturaleza. Pertenecientes a este grupo clasificamos los Algoritmos genéticos y otras técnicas más recientes, como la optimización por Enjambres de individuos (“Swarm Optimization”) [Kennedy y Eberhart, 1995] o la optimización por Colonia de hormigas (“Ant Colony Optimization”) [Dorigo et al., 2006].

Las técnicas locales están adaptadas a espacios de solución con características propias. Gracias a la información que toman del espacio de solución, la convergencia hacia un máximo local suele ser rápida. La mayor desventaja de esta clase de métodos radica en su dependencia del punto de inicio. Si dicho punto no es apropiado, la convergencia se verá degradada, y aumentará la probabilidad de caer en un máximo local no adecuado. El rendimiento de las técnicas globales no depende del punto inicial, y no toman información del espacio de solución. Su convergencia es así más lenta, pero son capaces de recorrer más ampliamente el espacio de solución de forma que encontrarán con mayor probabilidad un máximo global. En definitiva, sacrifican rapidez por independencia de las condiciones del problema de optimización. Dentro de las técnicas globales, aquellas basadas en mecanismos naturales gozan de unas ventajas propias. Las técnicas mencionadas parten de un conjunto de puntos iniciales y no de un punto en particular. La convergencia hasta una solución se halla en un punto intermedio entre la de las técnicas de gradiente y la del resto de técnicas globales. Por tanto su aplicación es deseable en aquellas situaciones en las que no se tenga un conocimiento profundo del espacio de solución. En este caso no es adecuado emplear técnicas de gradiente, ya que para calcular la derivada o derivadas de orden superior es necesario conocer si la función sobre la que se calcula dicho gradiente es continua o derivable hasta el orden preciso. Los AG pueden operar sobre funciones discontinuas o no diferenciables. El empleo de los AG es apropiado si desconocemos un punto inicial adecuado para un proceso de optimización vía gradiente. En ese caso los AG explorarán una región del espacio de solución suficientemente amplia para encontrar la solución deseada.

Los AG están basados en los mecanismos de la selección natural y de la genética [Goldberg, 1989]. Gracias a estos mecanismos, los individuos de una especie mejoran su capacidad de adaptación al medio. Ciertas características simplificadas de estos mecanismos son adaptadas en los Algoritmos genéticos para desarrollar un método de optimización. Así, en los AG se parte de una

población de individuos que intercambian material genético para generar una nueva población. En cada generación se busca que los nuevos individuos posean mejores características que sus antepasados. La mejora de la población se produce gracias a la selección como padres de aquellos individuos mejor adaptados. Para decidir qué individuos son los que ofrecen mejores prestaciones debe definirse una función de rendimiento. También existe la posibilidad de introducir nuevo material genético gracias a la mutación de los genes de un individuo. La mezcla de las características de los individuos padres junto con la mutación permite la exploración de nuevas zonas del espacio de solución. De esta forma, la sucesión ordenada de generaciones permite alcanzar un máximo (o de forma equivalente un mínimo) en un problema de optimización.

Los AG se han empleado en numerosos problemas electromagnéticos [Rahmat-Samii y Michielsen, 1999]. En la presente tesis se han empleado para implantar un nuevo método de diseño de filtros de microondas multicapa encapsulados. Se ha pretendido mejorar el punto débil de la técnica de AG mediante el desarrollo de una nueva función de rendimiento (“fitness”) que asegura una convergencia rápida y fiable. De esta forma, los AG podrán ser empleados en un número mayor de situaciones llegando a ser competitivos frente a las técnicas de gradiente. La función de rendimiento es un elemento fundamental en la operación del AG. La función de rendimiento debe expresar adecuadamente el grado de cumplimiento de las especificaciones por parte del filtro que representa un individuo de la población. Dicha función no solo debe expresar el mencionado grado de cumplimiento, sino que debe medirlo de forma que se evite el problema de la convergencia prematura. Este defecto aparece cuando un individuo obtiene una puntuación muy elevada en comparación con el resto de la población. Así, en la próxima generación todos los individuos hijos tendrán al individuo predominante como padre. Se pierde así la capacidad de explorar el espacio de solución ya que toda la población converge al individuo predominante. Como se mostrará en el Capítulo 5, la nueva función de rendimiento expresa el grado de satisfacción de las especificaciones, a la vez que evita los problemas de convergencia prematura. Estos beneficios son posibles gracias al novedoso uso de funciones de tipo logístico. Dichas funciones permiten la generación de un esquema de bonificación gradual, que depende del grado de cumplimiento de las especificaciones. Además, el uso de estas funciones posibilita la fijación de un máximo en la mencionada bonificación, limitando el rendimiento máximo de un circuito. Este hecho permite evitar los problemas de convergencia prematura. Gracias a la técnica neuronal de análisis en el medio multicapa encapsulado, el tiempo total empleado por el AG es admisible, incluso aunque implique la ejecución de varias generaciones completas. A pesar de la mejora en la convergencia efectuada, si el punto inicial es apropiado, una técnica de optimización por gradiente convergerá más rápidamente que el AG.

1.4.2. Técnica de diseño basada en redes neuronales y aproximación de parámetros eléctricos.

La segunda técnica desarrollada emplea redes neuronales para establecer las relaciones existentes entre los parámetros eléctricos de un circuito y las dimensiones físicas que los sintetizan. Las redes neuronales han sido empleadas en el diseño de circuitos de microondas de dos formas diferentes. En el primer método se emplean las redes neuronales utilizadas en el análisis del circuito que se desea diseñar. Estas redes toman como entradas las dimensiones físicas y devuelven la respues-

ta electromagnética del circuito. En este caso, las redes neuronales serán usadas de forma repetitiva hasta que se encuentren las dimensiones físicas que posibilitan un buen ajuste entre la respuesta buscada y la simulada. El segundo procedimiento se basa en el intercambio de las salidas y las entradas con respecto al problema de modelado de la respuesta electromagnética. De esta forma, las entradas de la red neuronal pasan a ser las dimensiones físicas y las salidas los parámetros eléctricos [Wang et al., 2006, Kabir et al., 2008]. A esta última metodología se le denomina diseño inverso de forma directa. El segundo método de diseño desarrollado sigue esta última filosofía y se aplica por primera vez al diseño de varios filtros multicapa (microtira) encapsulados.

Uno de los métodos más extendidos para diseñar un filtro de microondas descansa en la obtención de una respuesta ideal de tipo Butterworth o Chebychev. Esta respuesta se selecciona junto con el orden del filtro de forma que el prototipo equivalente paso bajo satisface las especificaciones establecidas. Si el filtro está basado en una sucesión de resonadores, los factores de calidad y coeficientes de acoplo de dichos resonadores se pueden calcular a partir de los coeficientes del prototipo paso bajo. El problema fundamental radica en la obtención de las dimensiones de los resonadores que sintetizan los coeficientes de acoplo y factores de calidad calculados analíticamente. Una solución consiste en la generación de tablas y curvas que relacionen las dimensiones físicas de los resonadores con las parámetros eléctricos [Hong y Lancaster, 2001]. Como ya se comentó en el principio de esta sección, estas curvas de nivel necesitan la presencia de un ingeniero, no permiten la automatización del proceso de diseño, y requieren numerosos datos para lograr cierta precisión. La solución propuesta en la presente tesis consiste en la utilización de redes neuronales. Las raíces del método se encuentran en [Wang et al., 2006, Kabir et al., 2008], trabajos en los que el MLP se empleó para calcular las dimensiones físicas que sintetizaban ciertos elementos de acoplo de una matriz de acoplos perteneciente a un filtro en guíaonda. La novedad añadida en el presente trabajo es el empleo en filtros impresos en un medio multicapa apantallado, y del uso de las RBFNNs como red neuronal. Como se mostrará en el Capítulo 3, las RBFNNs aproximan con gran precisión y economía de datos las relaciones entre dimensiones físicas y variaciones de los parámetros eléctricos. Al igual que en el método de diseño con AG, el nuevo método neuronal de análisis es empleado como simulador electromagnético. Gracias a la herramienta neuronal, el periodo de generación de los conjuntos de entrenamiento posee una duración pequeña, de modo que que posibilita la aplicación práctica del método.

1.5. Estructura de la Tesis

La memoria de la tesis sigue la siguiente estructura dividida en capítulos:

- Funciones de Green del medio multicapa apantallado: en este capítulo se explica brevemente la formulación empleada para calcular las funciones de Green de los potenciales del medio multicapa apantallado. Asimismo se muestran los problemas típicos del cálculo de dichas funciones como la lenta convergencia de las series asociadas.
- Método de análisis neuronal basado en RBFNNs: en este capítulo se desarrolla el método de análisis neuronal de circuitos impresos multicapa y encapsulados. Se justifica la elección de las RBFNNs como red neuronal, así como la selección de un método de entrenamiento apropiado para la tarea de aproximación. Se implementan una serie de algoritmos de división del espacio

de entrada para alcanzar una correcta aproximación de las funciones de Green. Dicho proceso de división del espacio de entrada se complementa con un mecanismo que permite calcular el valor apropiado de varianza gaussiana en las neuronas que componen las RBFNNs. El capítulo finaliza con la aplicación de una técnica novedosa de extracción de la singularidad en las funciones de Green, que posibilita un drástico aumento de la precisión. Cada una de las técnicas desarrolladas se aplica al análisis de circuitos prácticos.

- **Método de análisis neuronal basado en Redes neuronales de Chebychev:** en este capítulo se desarrolla una nueva red neuronal fundamentada en polinomios de Chebychev. En primer lugar la nueva red neuronal se aplica a funciones matemáticas para demostrar su validez en tareas de regresión. A continuación se exhibe la capacidad de la nueva red neuronal en la aproximación de las funciones de Green. Para analizar los circuitos impresos encapsulados se utilizan los algoritmos de división del espacio de entrada expuestos en el capítulo anterior. Los resultados muestran que el método de entrenamiento aplicado a la nueva red neuronal permite una aceleración del análisis mucho mayor que la alcanzada con las RBFNNs con un nivel de precisión similar.
- **Métodos de diseño de circuitos de microondas encapsulados multicapa:** en este capítulo se desarrollan los dos procedimientos de diseño. En primer lugar se muestra la técnica de diseño de circuitos basada en AG. Se implementa una nueva función de rendimiento que permite una convergencia sólida y rápida hacia una solución que satisface las especificaciones establecidas. Esta función de rendimiento evita la aparición del problema de la convergencia prematura. Finalmente, se desarrolla el método de diseño en el que las RBFNNs aproximan las relaciones entre los factores de calidad y coeficientes de acoplo con las dimensiones físicas que los sintetizan. En primer lugar se explica el procedimiento para calcular los factores de calidad y coeficientes de acoplo necesarios para completar los conjuntos de entrenamiento y validación. Una vez entrenadas las RBFNNs, se comprueba que las redes neuronales son capaces de calcular las dimensiones físicas adecuadas. Para validar la técnica se muestran diferentes ejemplos prácticos de filtros impresos apantallados.
- **Conclusiones y líneas futuras:** en este capítulo se muestran las conclusiones sobre los métodos de análisis y diseño desarrollados en la tesis. Además, se exponen diversas consideraciones sobre las mejoras a introducir en los algoritmos implementados, y las extensiones que se pueden realizar del trabajo realizado.
- **Apéndices:** en los apéndices se exhiben aquellos desarrollos matemáticos que por su extensión entorpecen la lectura de la tesis. Por la misma razón, también se recopilan resultados prácticos fruto de la aplicación del método neuronal.
- **Índice alfabético y glosario:** finalmente en estos apartados se recogen los términos más importantes utilizados a lo largo de todos los capítulos. El índice alfabético recoge las páginas en las que aparecen los términos más importantes. El glosario muestra una breve definición de estos términos. En la presente tesis se han seleccionado una serie de símbolos y siglas que prácticamente constituyen un estándar en la literatura científica de los campos tratados. En la parte concerniente a la descripción de las funciones de Green y diseño de filtros encapsulados se han utilizado símbolos ampliamente representados en libros y trabajos de referencia del campo

del electromagnetismo [Collin, 1960], [Collin, 1966], [Matthaei et al., 1980], [Marcuvitz, 1951], [Pozar, 1998] y [Hong y Lancaster, 2001]. Asimismo, se ha intentado usar aquellos símbolos más comunes en las técnicas de inteligencia artificial como las redes neuronales [Haykin, 1999] y los Algoritmos genéticos [Goldberg, 1989]. Debido a que la mayor parte de la literatura científica se encuentra escrita en lengua inglesa se ha preferido, en general, la utilización de siglas en este idioma que son ampliamente conocidas en la comunidad científica hispanohablante. En algún caso se ha empleado siglas en español para contribuir a su difusión. También se han utilizado subíndices que aluden a términos en español. Las listas de figuras y tablas concluyen la tesis.

1.6. Contribuciones originales de la presente tesis

La presente tesis ha producido las siguientes contribuciones originales:

- Método de análisis neuronal basado en RBFNN: en primer lugar se comprobó la validez de las RBFNNs en la aproximación del módulo de las funciones de Green del medio multicapa. Esta primera aproximación al problema produjo las comunicaciones orales [Garcia et al., 2004a, Garcia et al., 2004d] y el artículo en revista científica [Garcia et al., 2004c]. Para que las redes neuronales sustituyan los valores de las funciones de Green en la resolución de la IE, es necesario que sean capaces de aproximar la parte real e imaginaria de las mencionadas funciones de Green. La demostración de las habilidades de las RBFNNs en la aproximación la parte real e imaginaria de las funciones de Green, tuvo como fruto la comunicación en forma de panel en congreso internacional [Garcia et al., 2004b]. A continuación se desarrolló una serie de algoritmos para la aproximación completa de las funciones de Green del medio apantallado. Las redes neuronales entrenadas fueron incorporadas en el software de análisis de circuitos impresos mediante la IE. Los resultados obtenidos y los algoritmos implementados produjeron las comunicaciones orales [Garcia et al., 2005a, Garcia et al., 2005b] y la publicación del artículo en revista científica [Garcia et al., 2006b]. Para aumentar la precisión y rapidez del método neuronal se desarrolló un nuevo método de eliminación de la singularidad basado en la extracción de dos capas de imágenes espaciales. Además, se generaron nuevos tipos de zonas circulares, adecuadas para expresar las variaciones de las funciones de Green, cerca de la fuente. Estas nuevas técnicas, especialmente el novedoso procedimiento de cancelación de la singularidad, posibilitan un drástico incremento de la precisión del análisis.
- Método de análisis neuronal basado en redes neuronales de Chebychev: se desarrolló un nuevo tipo de red neuronal basado en polinomios de Chebychev. La nueva red neuronal, aunque similar a otras redes basadas en polinomios de Chebychev, emplea un método de entrenamiento nunca utilizado previamente en ningún tipo de redes neuronales fundamentadas en polinomios. Este método de entrenamiento se implantó originalmente para diseñar RBFNNs. Sin embargo, no alcanza un rendimiento óptimo en esta clase de redes neuronales. En las nuevas redes de Chebychev, el mencionado método de entrenamiento consigue un resultado óptimo. De esta forma, se consiguen redes neuronales mucho más compactas en mucho menos tiempo. Como consecuencia de estos beneficios, el análisis neuronal es más veloz que el conseguido con RBFNNs, conservando la precisión conseguida con estas redes neuronales. Las nuevas re-

des de Chebychev utilizan la nueva técnica de extracción de imágenes espaciales para eliminar la singularidad.

- El método de diseño basado en AG produjo la comunicación oral en congreso internacional [Garcia et al., 2006a] y la comunicación oral en congreso nacional [Garcia et al., 2006c]. Finalmente, el segundo método de diseño inverso directo, basado en redes neuronales, ha tenido como fruto la comunicación oral en congreso nacional [Garcia et al., 2007], y la publicación del artículo en revista científica [Garcia et al., 2009].

Capítulo 2

Evaluación de las funciones de Green del medio encapsulado multicapa

2.1. Introducción

En este Capítulo se detallan los algoritmos empleados para calcular las funciones de Green del medio multicapa apantallado. En esta tesis el método neuronal desarrollado se ha limitado al estudio de estructuras multicapa totalmente encapsuladas en las que las únicas fuentes son de tipo eléctrico. Por tanto, el desarrollo de las funciones de Green está restringido a excitaciones de tipo eléctrico quedando excluidas así el uso de excitaciones de tipo magnético que se utilizan para modelar algunos elementos como ranuras.

El mayor impedimento en la aplicación de la Ecuación integral (IE) en problemas prácticos radica en la lentitud de la evaluación de las funciones de Green. Existen diversos enfoques para orillar este problema. La IE se puede formular en el dominio espectral, de modo que los términos de las matriz de impedancias generalizadas dependen de las sumas infinitas que forman las funciones de Green. Estos términos pueden ser evaluados de forma eficiente mediante la FFT [Hill y Tripathi, 1991]. En general, las formulaciones espectrales presentan problemas de convergencia cuando la cavidad presenta unas dimensiones mucho mayores que el circuito impreso. En las formulaciones espaciales las funciones de Green son evaluadas cada vez que se utilizan en la resolución de la IE. Gracias a este hecho, dichas formulaciones no presentan el mencionado inconveniente. No obstante, en este dominio las funciones de Green están compuestas por series infinitas de convergencia lenta. Una de las técnicas más utilizadas para acelerar el cálculo de las funciones de Green en el dominio espacial es el método de Kummer consistente en la separación de la parte estática y dinámica de las series asociadas. La parte estática se puede, o bien calcular de forma analítica, o bien expresar como una suma de imágenes espaciales de rápida convergencia cerca de la fuente mediante la fórmula de Poisson. En la presente tesis se han aplicado de forma combinada un procedimiento como el anterior, basado en teoría de imágenes espaciales, y un método de suma de series en el dominio espectral para calcular eficazmente las funciones de Green del medio.

La evaluación de las funciones de Green consta de diferentes pasos y procedimientos que deben

ser combinados ordenadamente. Como en todo problema electromagnético se parte de las ecuaciones de Maxwell y unas ciertas condiciones de contorno. En primer lugar las ecuaciones de Maxwell se separan en una parte longitudinal y otra parte transversal. Suponemos que la estructura se encuentra orientada de forma que la parte longitudinal posee únicamente la componente \hat{z} mientras que la parte transversal tiene componentes \hat{x} e \hat{y} . La parte transversal se resuelve aplicando una representación basada en funciones modales [Marcuvitz, 1951]. La representación mediante funciones modales constituye una expansión en series de Fourier, que permite expresar la parte longitudinal de los campos en forma de ecuaciones de una línea de transmisión. Por tanto, la parte longitudinal se calculará a partir de los voltajes y corrientes definidos en un circuito equivalente en línea de transmisión a la estructura de trabajo. Dicho circuito está formado por líneas de transmisión interconectadas, ver [Pan y Wolff, 1994a] y [Michalski y Mosig, 1997]. Una vez obtenidos los voltajes y corrientes de la parte longitudinal, se puede formar la serie completa basada en la representación modal. Tal y como se demuestra en [Marcuvitz, 1951] las funciones modales están compuestas por funciones sinusoidales. Una serie constituida por este tipo de funciones tiene un comportamiento oscilante por lo que serán necesarios un gran cantidad de términos para alcanzar cierta precisión en la suma resultante.

La solución propuesta tomada en este trabajo para calcular las series modales radica en la aplicación de la fórmula de Poisson [Collin, 1960] a la expansión modal, de modo que el desarrollo en series de Fourier original se transforma en una serie de términos en el dominio espacial. Cada término de la serie espacial equivale a una imagen de la fuente original desplazada en los ejes x e y . Gracias a la suma de imágenes, el número de términos necesarios para alcanzar la precisión deseada en la evaluación de las funciones de Green se ve reducido ampliamente cerca de la fuente. Sin embargo, este procedimiento presenta problemas de convergencia cuando aumenta la frecuencia debido a la excitación de ondas de superficie, y cuando el punto de observación está lejos de la fuente. Por consiguiente, el método de la suma de imágenes se aplicará únicamente a la parte cuasi-estática en la que no aparecen esta clase de ondas, y cuando el punto de observación se ubica cerca de la fuente. La serie original una vez extraída la parte cuasi-estática converge rápidamente, de forma que se puede sumar directamente. Por ello se denomina a este procedimiento método híbrido.

Hay que tener en cuenta que la suma de imágenes espaciales presenta una convergencia rápida sobre todo en aquellas situaciones en las que el punto observación se encuentra cerca de la fuente. Por el contrario si el punto observación se sitúa lejos de la fuente el método converge muy lentamente. Debido a esta razón, para aquellos puntos observación localizados lejos de la fuente se aplica un método fundamentado en la suma por partes de la serie original formulada en el dominio espectral [Melcon y Mosig, 2000]. Ahora, cada término de la suma está formado por sumas parciales de la serie original. Se observa que este procedimiento converge lentamente cerca de la fuente, debido a la singularidad de las funciones de Green, mientras que acelera la evaluación de las funciones de Green lejos de la fuente. Por esta razón el espacio transversal se va a dividir en dos zonas. En la zona cercana a la fuente se empleará el método híbrido mientras que en la zona lejana se utilizará la técnica de la suma por partes.

En este Capítulo se van a mostrar los fundamentos de las técnicas empleadas ya que el objetivo de la tesis no es la mejora de los algoritmos de cálculo de las funciones de Green sino su sustitución por un conjunto de redes neuronales. Una vez entrenadas correctamente calcularán los valores nece-

sarios durante la resolución de la IE de forma mucho más veloz que las técnicas de suma de series explicados en este Capítulo. Una visión más profunda de los algoritmos empleados se encuentra en [Álvarez Melcón, 1998]. La estructura de este Capítulo es la siguiente. En el apartado 2.2 se describe el modelo circuital de la estructura multicapa apantallada que se obtiene al aplicar la formulación modal a la parte transversal de las funciones de Green. El modelo circuital permite calcular la parte longitudinal empleando la teoría de líneas de transmisión. Al final del apartado se presenta la formulación general de las funciones de Green de los potenciales del medio. En el apartado 2.3 se muestra el procedimiento seguido para calcular las series modales cuando el punto observación se ubica cerca de la fuente basado en la suma de imágenes espaciales. A continuación, en el apartado 2.4, se expone la técnica empleada para la suma de las series modales del dominio espectral cuando el punto observación se encuentra lejos de la fuente. En este apartado se muestra la metodología aplicada para determinar el límite que separa la zona de puntos cercanos a la fuente de la zona que incluye a los puntos observación situados lejos de la fuente. En el apartado 2.5 se hace hincapié en ciertos aspectos de las funciones de Green que van a tener una influencia decisiva en su aproximación por medio de las redes neuronales. Las funciones de Green en una cavidad rectangular presentan simetrías que serán explotadas en el entrenamiento de las redes neuronales. Asimismo se ofrecen varios ejemplos que muestran las posibles dificultades en la tarea de la aproximación de las funciones de Green por parte de las redes neuronales. Finalmente, en las conclusiones se realiza un breve resumen de lo expuesto en el presente Capítulo.

2.2. Formulación de las funciones de Green de los potenciales del medio multicapa apantallado

La estructura multicapa apantallada estará compuesta por varias capas de material dieléctrico en el que se imprimirán zonas con material conductor. Las capas de dieléctrico estarán insertadas en una cavidad metálica en forma de hexaedro. Las zonas conductoras serán excitadas desde el exterior de forma que se inducirán una serie de corrientes eléctricas en el interior de la cavidad. Por lo tanto, esta estructura puede considerarse como una guíaonda rellena por diversas capas de distintos materiales dieléctricos y finalizada por paredes metálicas. En [Michalski y Mosig, 1997] y [Pan y Wolff, 1994b] se demuestra que las funciones de Green de los potenciales (G_{cV} , G_{cA}^{xx} , G_{cA}^{yy}) de la cavidad multicapa poseen la siguiente expresión basada en una expansión mediante funciones modales propias de la estructura bajo estudio:

$$G_{cV} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \tilde{G}_V(k_{x_m}, k_{y_n}) \sin(k_{x_m}x) \sin(k_{x_m}x') \sin(k_{y_n}y) \sin(k_{y_n}y') \quad (2.1)$$

$$G_{cA}^{xx} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \tilde{G}_A^{xx}(k_{x_m}, k_{y_n}) \cos(k_{x_m}x) \cos(k_{x_m}x') \sin(k_{y_n}y) \sin(k_{y_n}y') \quad (2.2)$$

$$G_{cA}^{yy} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \tilde{G}_A^{yy}(k_{x_m}, k_{y_n}) \sin(k_{x_m}x) \sin(k_{x_m}x') \cos(k_{y_n}y) \cos(k_{y_n}y') \quad (2.3)$$

donde k_{x_m} y k_{y_n} son los números de onda transversales a lo largo de los ejes x e y respectivamente que toman los siguientes valores:

$$k_{x_m} = \frac{m\pi}{a} \quad k_{y_n} = \frac{n\pi}{b} \quad (2.4)$$

donde a y b son las dimensiones transversales de la cavidad como se aprecia en la Figura 2.1. Las funciones modales formadas por sinusoides permiten expresar las funciones de Green como un desarrollo en serie de Fourier respecto a las coordenadas transversales (x, y) . Así, las funciones en el dominio espectral $\tilde{G}_V, \tilde{G}_A^{xx}$, y \tilde{G}_A^{yy} corresponden a los coeficientes de dicha serie Fourier. Las funciones modales se pueden agrupar en dos funciones f y g que dependen bien de (k_{x_m}, x, x') o bien de (k_{y_n}, y, y') , tal como se observa en la Tabla 2.1. Las funciones f y g serán de utilidad cuando se aplique la técnica de suma por partes a las series modales en el dominio espectral.

Tabla 2.1: Expresión de las funciones f y g para las diferentes funciones de Green de los potenciales.

	$f(k_{x_m}, x, x')$	$g(k_{y_n}, y, y')$
G_{cV}	$\sin(k_{x_m}x) \sin(k_{x_m}x')$	$\sin(k_{y_n}y) \sin(k_{y_n}y')$
G_{cA}^{xx}	$\cos(k_{x_m}x) \cos(k_{x_m}x')$	$\sin(k_{y_n}y) \sin(k_{y_n}y')$
G_{cA}^{yy}	$\sin(k_{x_m}x) \sin(k_{x_m}x')$	$\cos(k_{y_n}y) \cos(k_{y_n}y')$

Las funciones del dominio espectral se pueden calcular empleando el modelo circuital equivalente a la estructura multicapa que se muestra en la Figura 2.1. Dicho modelo circuital proviene de la escalarización de las ecuaciones de Maxwell que describen el comportamiento de los campos electromagnéticos. Esta escalarización permite la separación de las variaciones longitudinales respecto de las componentes transversales. Según el modo sea transversal eléctrico (TE) o transversal magnético (TM) se obtendrán diferentes expresiones. Para cada uno de los dos tipos de modos los coeficientes de la variación longitudinal del campo eléctrico y magnético se pueden calcular mediante una serie de ecuaciones que se asemejan a las ecuaciones de una línea de transmisión. Los coeficientes mencionados serán por tanto los voltajes y corrientes del modelo circuital equivalente. Dicho modelo circuital equivalente posee una estructura similar para todos los modos $TE_{m,n}$ y $TM_{m,n}$ con la diferencia de que cada línea de transmisión tiene una impedancia característica propia del modo correspondiente. Para el caso de una excitación de tipo eléctrico, situada con coordenada z' (por tanto entre la interfaz de los medios dieléctricos u y $u - 1$ como se aprecia en la Figura 2.1), un modo con índice (m, n) y un punto observación situado en z , la expresión de las funciones de Green espectrales toma el valor para cada potencial:

$$\tilde{G}_{cV} = j\omega V_{m,n}(z, z') N_{b_{m,n}} \quad (2.5)$$

$$\tilde{G}_{cA}^{xx} = + \frac{1}{j\omega} V_{m,n}(z, z') N_{a_{m,n}} \quad (2.6)$$

$$\tilde{G}_{cA}^{yy} = -\frac{1}{j\omega} V_{m,n}(z, z') N_{a_{m,n}} \quad (2.7)$$

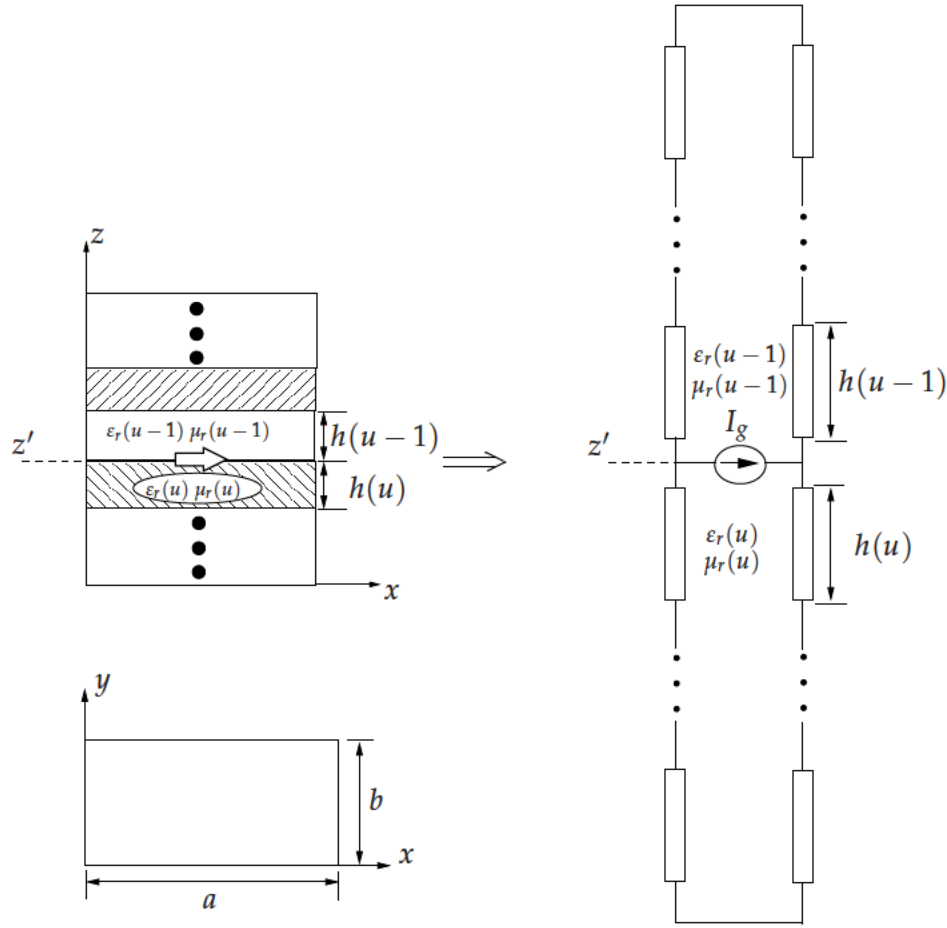


Figura 2.1: Dipolo elemental radiante insertado en una estructura multicapa apantallada y su modelo circuital equivalente.

donde $V_{m,n}(z, z')$ es el voltaje en la coordenada z que produce el generador de intensidad que representa la excitación puntual colocada en z' . Los términos $N_{b_{m,n}}$ y $N_{a_{m,n}}$ son factores de normalización empleados para que la expansión en funciones modales forme un conjunto ortonormal. El voltaje $V_{m,n}(z, z')$ tomará diferentes valores dependiendo de si el modo es TE o TM. Este hecho es debido a que el voltaje depende de las impedancias características de cada línea, y éstas dependen a su vez del modo $TE_{m,n}$ o $TM_{m,n}$. Los factores de normalización y las impedancias características toman los siguientes valores:

$$N_{b_{m,n}} = \frac{\epsilon_m \epsilon_n}{\pi^2 k_{\rho_{m,n}}^2} \quad \epsilon_m = \begin{cases} 1 & \text{si } m = 0 \\ 2 & \text{si } m \neq 0 \end{cases} \quad k_{\rho_{m,n}} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \quad (2.8)$$

$$N_{a_{m,n}}^{TE} = \frac{\epsilon_m \epsilon_n}{a b} \quad N_{a_{m,n}}^{TM} = 0 \quad \epsilon_m = \begin{cases} 1 & \text{si } m = 0 \\ 2 & \text{si } m \neq 0 \end{cases} \quad (2.9)$$

$$Z_{c_{m,n}}^{TE}(u) = \frac{\omega\mu_0\mu_r(u)}{\beta_{m,n}^{(u)}} \quad Z_{c_{m,n}}^{TM}(u) = \frac{\beta_{m,n}^{(u)}}{\omega\epsilon_0\epsilon_r(u)} \quad \beta_{m,n}^{(u)} = \sqrt{k_0^2\epsilon_r(u)\mu_r(u) - k_{p_{m,n}}^2} \quad (2.10)$$

donde $\beta_{m,n}^{(u)}$ es la constante de propagación en la línea de transmisión correspondiente a la capa u y ϵ_m es el factor de Neumann (no confundir con la permitividad). Se observa que el factor del potencial eléctrico escalar $N_{b_{m,n}}$ toma el mismo valor tanto si el modo es TE como TM, mientras que el valor del factor $N_{a_{m,n}}$ del potencial magnético vectorial depende del modo en cuestión.

Para calcular el voltaje producido por el generador de intensidad entre los medios (i) e $(i - 1)$ se recurre al análisis del modelo equivalente de la Figura 2.2. El índice i recorre todas las capas desde b hasta t como se observa en la Figura 2.2. Cada línea de transmisión equivalente poseerá su propia constante de propagación $\beta_{m,n}^i$ e impedancia característica $Z_{c_{m,n}}^p(i)$ donde p puede tomar los valores TE, TM. Para simplificar la notación eliminamos el subíndice compuesto m, n y el superíndice p de forma que la impedancia característica de la capa i -ésima pasa a ser $Z_c(i)$. El modelo circuital está limitado tanto por arriba como por debajo del generador por sendos cortocircuitos que representan las tapas metálicas que cierran la cavidad. La primera capa de la estructura, situada en la parte superior, se indica con el subíndice t y la última capa, en la parte inferior, se indica con el subíndice b . Para calcular los voltajes necesarios en primer lugar se transforma cada línea de transmisión en una red de impedancias de dos puertos tal como se observa en la Figura 2.3. Las impedancias de dicha red de dos puertos son:

$$Z_a(i) = jZ_{c_{m,n}}^p(i) \tan(\beta_{m,n}^i l_i / 2) \quad (2.11a)$$

$$Z_b(i) = -jZ_{c_{m,n}}^p(i) \csc(\beta_{m,n}^i l_i) \quad (2.11b)$$

$$Z_{11}(i) = Z_{22}(i) = Z_a(i) + Z_b(i) \quad (2.11c)$$

$$Z_{12}(i) = Z_{21}(i) = Z_b(i) \quad (2.11d)$$

De esta forma el voltaje de la red entre las líneas (i) e $(i - 1)$ situadas por encima del generador de corriente se calcula como:

$$I_1(i) = \frac{V_2(i) - Z_{22}(i)I_2(i)}{Z_{21}(i)} \quad V_2(i) = V_1(i + 1) \quad (2.12a)$$

$$V_1(i) = Z_{11}(i)I_1(i) + Z_{12}(i)I_2(i) \quad I_2(i) = -I_1(i + 1) \quad i = u - 1, u - 2, \dots, t \quad (2.12b)$$

Mientras que el voltaje de la red entre las líneas i e $i + 1$ situadas por debajo del generador de corriente toma la expresión:

$$I_2(i) = \frac{V_1(i) - Z_{11}(i)I_1(i)}{Z_{12}(i)} \quad V_1(i) = V_2(i - 1) \quad (2.13a)$$

$$V_2(i) = Z_{21}(i)I_1(i) + Z_{22}(i)I_2(i) \quad I_1(i) = -I_2(i - 1) \quad i = u, u + 1, \dots, b \quad (2.13b)$$

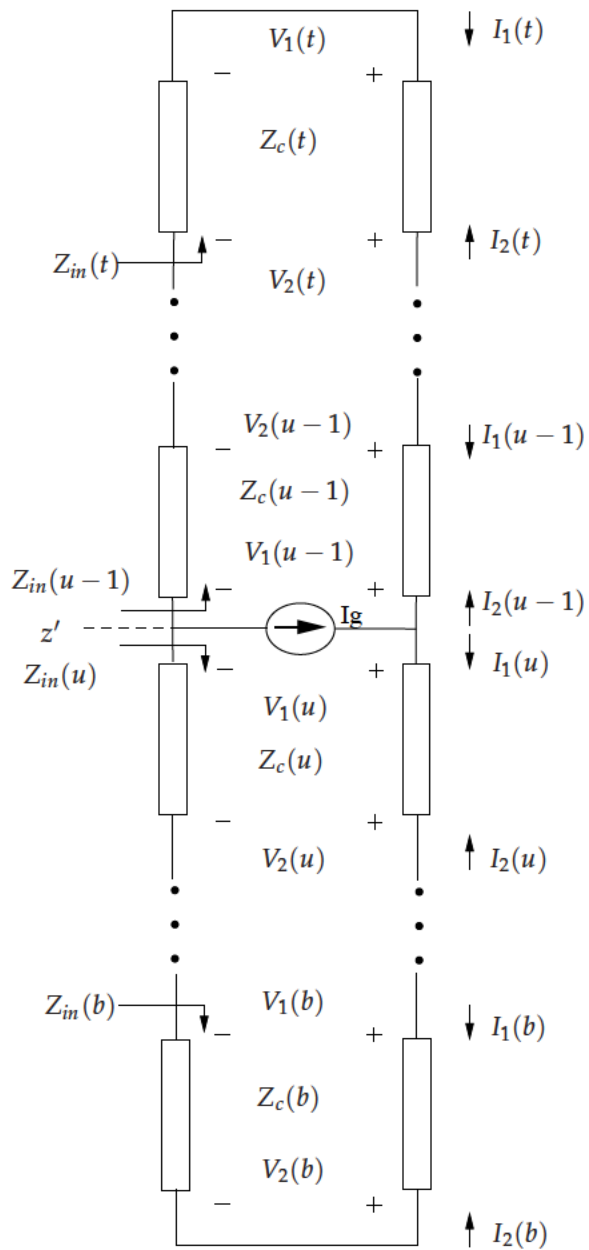


Figura 2.2: Voltaje, intensidades e impedancias de entrada del modelo circuital equivalente. Para facilitar la comprensión del esquema se ha omitido la dependencia con el índice del modo (m, n) y el tipo de modo (TE, TM)

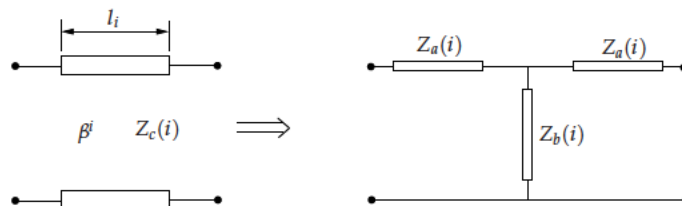


Figura 2.3: Red de impedancias equivalente de una línea de transmisión.

Los valores de tensión y corriente desconocidos ($V_1^{(i+1)}, I_1^{(i+1)}, V_2^{(i-1)}, I_2^{(i-1)}$) en las ecuaciones (2.12) y (2.13) se calculan de forma iterativa a partir de las tensiones y corrientes que aparecen en la interfaz de las capas u y $u - 1$:

$$V_2(u-1) = V_1(u) = I_g \frac{Z_{in}(u-1)Z_{in}(u)}{Z_{in}(u-1) + Z_{in}(u)} \quad (2.14a)$$

$$I_2(u-1) = \frac{V_2(u-1)}{Z_{in}(u-1)} \quad (2.14b)$$

$$I_1(u) = \frac{V_1(u)}{Z_{in}(u)} \quad (2.14c)$$

El cómputo de los voltajes permite construir, junto con los factores de normalización y las funciones modales apropiadas, la serie que forma cada una de las funciones de Green de los potenciales del medio. La razón para elegir las funciones de Green de los potenciales en lugar de las funciones de Green de los campos E y H es debido a que las primeras presentan singularidades de menor orden que las funciones de Green de los campos electromagnéticos. La formulación de la IE que usa las funciones de Green de los potenciales se denomina Ecuación integral de los Potenciales Mixtos ("Mixed Potential Integral Equation" o MPIE según las siglas inglesas). Esta formulación es la utilizada a lo largo de toda la tesis, a partir de ahora, por simplicidad, nos referiremos a ella como la IE. A pesar de la mencionada ventaja numérica, las funciones de Green asociadas a los potenciales escalar y vectorial presentan una convergencia muy lenta en caso de aplicar una suma directa. Como ya se comentó, este inconveniente es causado por la dependencia por parte de la serie de funciones sinusoidales oscilantes. A continuación describimos las técnicas empleadas en este trabajo para obtener los valores de las funciones de Green de una forma más adecuada respecto al caso de la suma directa.

2.3. Método híbrido

Una solución para acelerar el cálculo de las funciones de Green de los potenciales consiste en aplicar la fórmula de la suma de Poisson [Collin, 1960]:

$$\sum_{n=-\infty}^{\infty} G(\alpha n) = \frac{\sqrt{2\pi}}{\alpha} \sum_{n=-\infty}^{\infty} \tilde{G}\left(\frac{2n\pi}{\alpha}\right) \quad (2.15)$$

donde \tilde{G} representa la transformada de Fourier de la función G . Si aplicamos la fórmula anterior a una función bidimensional definida en el dominio espacial y desplazada tanto en x como en y obtenemos:

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} G(x+2ma, y+2nb) = \frac{\pi}{2ab} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \tilde{G}(k_{x_m}, k_{y_n}) e^{jk_{x_m}x} e^{jk_{y_n}y} \quad (2.16)$$

La ecuación anterior se puede emplear para transformar las series basadas en funciones sinusoidales de las ecuaciones (2.1,2.2,2.3), en series compuestas por versiones desplazadas de la función

de Green G correspondiente en el dominio espacial. Este tipo de funciones presentan un mejor comportamiento desde el punto de vista numérico cerca de la fuente. Vamos a aplicar la ecuación (2.16) a la función de Green del potencial eléctrico escalar para obtener una nueva expresión basada en la función de Green G correspondiente. Debido a que G_{cV} depende únicamente de funciones seno podemos escribir la ecuación (2.1) como:

$$G_{cV} = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \tilde{G}_V(k_{x_m}, k_{y_n}) \sin(k_{x_m} x) \sin(k_{x_m} x') \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.17)$$

Para poder aplicar la fórmula (2.16) hay que expresar las funciones sinusoidales en forma exponencial:

$$G_{cV} = -\frac{1}{4} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \tilde{G}_V(k_{x_m}, k_{y_n}) \sin(k_{x_m} x) \sin(k_{y_n} y) \left[e^{+jk_{x_m} x'} e^{+jk_{y_n} y'} - e^{+jk_{x_m} x'} e^{-jk_{y_n} y'} - e^{-jk_{x_m} x'} e^{+jk_{y_n} y'} + e^{-jk_{x_m} x'} e^{-jk_{y_n} y'} \right] \quad (2.18)$$

A continuación extendemos el sumatorio desde $-\infty$ hasta $+\infty$, para conseguirlo se realiza la siguiente operación:

$$\begin{aligned} \sum_{m=1}^{\infty} \sin(k_{x_m} x) &= \sum_{m=1}^{\infty} \frac{e^{+jk_{x_m} x} - e^{-jk_{x_m} x}}{2j} \\ &= \frac{1}{2j} \left[\sum_{m=+1}^{\infty} e^{+jk_{x_m} x} - \sum_{m=-1}^{-\infty} e^{+jk_{x_m} x} \right] \\ &= \frac{1}{2j} \sum_{m=-\infty}^{\infty} \text{signo}(m) e^{+jk_{x_m} x} \end{aligned} \quad (2.19)$$

Aplicando dos veces la ecuación (2.19) a la ecuación (2.18) obtenemos el siguiente desarrollo:

$$G_{cV} = \frac{1}{16} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \tilde{G}_V(k_{x_m}, k_{y_n}) e^{+jk_{x_m} x} e^{+jk_{y_n} y} \left[e^{+jk_{x_m} x'} e^{+jk_{y_n} y'} - e^{+jk_{x_m} x'} e^{-jk_{y_n} y'} - e^{-jk_{x_m} x'} e^{+jk_{y_n} y'} + e^{-jk_{x_m} x'} e^{-jk_{y_n} y'} \right] \quad (2.20)$$

La ecuación anterior se reescribe de forma que los exponentes queden agrupados como:

$$\begin{aligned} G_{cV} &= \frac{1}{16} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \tilde{G}_V(k_{x_m}, k_{y_n}) \\ &\quad \left\{ e^{[+jk_{x_m}(x+x')] + [jk_{y_n}(y+y')]} - e^{[+jk_{x_m}(x+x')] + [jk_{y_n}(y-y')]} \right. \\ &\quad \left. - e^{[+jk_{x_m}(x-x')] + [jk_{y_n}(y+y')]} + e^{[+jk_{x_m}(x-x')] + [jk_{y_n}(y-y')]} \right\} \end{aligned} \quad (2.21)$$

Finalmente aplicamos la transformación de Poisson a la ecuación anterior para obtener:

$$G_{cV} = \frac{ab}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [G_V(x+x'+2ma, y+y'+2nb) - G_V(x+x'+2ma, y-y'+2nb) - G_V(x-x'+2ma, y+y'+2nb) + G_V(x-x'+2ma, y-y'+2nb)] \quad (2.22)$$

donde G_V es la función de Green definida en el dominio espacial producida al realizar la transformada inversa de Fourier de la función \tilde{G}_V . La ecuación (2.22) es consecuencia de la aplicación a la cavidad metálica de la teoría de imágenes. Dicha teoría indica que las paredes metálicas laterales son sustituidas por el conjunto de imágenes espaciales. De esta forma el potencial eléctrico escalar generado por una fuente puntual dentro de la cavidad se expresa mediante la suma de los potenciales eléctricos escalares de la mencionada serie infinita de imágenes espaciales. Los potenciales de la serie constituyen de esta forma un conjunto de funciones de Green de espacio libre periódicas. Además, G_V puede ser calculado a partir de \tilde{G}_V gracias al empleo de la integral de Sommerfeld correspondiente al caso de una estructura multicapa de dimensiones transversales infinitas, es decir, desposeída de las paredes metálicas. El procedimiento anterior se puede aplicar al resto de funciones de Green, de forma que se genera una serie de imágenes similar. De este modo, la función de Green de cualquier potencial se expresa como:

$$G_c(x, y, x', y') = \frac{ab}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [G(x+x'+2ma, y+y'+2nb) + s_x G(x-x'+2ma, y+y'+2nb) + s_y G(x+x'+2ma, y-y'+2nb) + s_x s_y G(x-x'+2ma, y-y'+2nb)] \quad (2.23)$$

donde G es la correspondiente función de Green del dominio espacial y s_x y s_y son funciones signo que toman diferentes valores dependiendo de la función de Green considerada. El valor de las funciones signo se observa en la Tabla 2.2:

Tabla 2.2: Valor de las funciones signo empleadas para definir las funciones de Green apantalladas

-	s_x	s_y
G_{cV}	-1	-1
G_{cA}^{xx}	+1	-1
G_{cA}^{yy}	-1	+1

La ecuación (2.23) representa un conjunto de cuatro imágenes situado en $(+x', +y')$, $(-x', +y')$, $(+x', -y')$ y $(-x', -y')$ respectivamente, que es desplazado en el espacio infinitas veces. En el eje x el desplazamiento depende una distancia igual a $2ma$, mientras que en el eje y el desplazamiento depende de la cantidad $2nb$. A cada grupo de cuatro imágenes lo denominaremos como conjunto de imágenes básico. En la Figura 2.4 podemos apreciar distintos conjuntos básicos en torno a la cavidad

metálica. La adecuada disposición en el espacio de las imágenes permite el cumplimiento de las condiciones de contorno de los potenciales del medio.

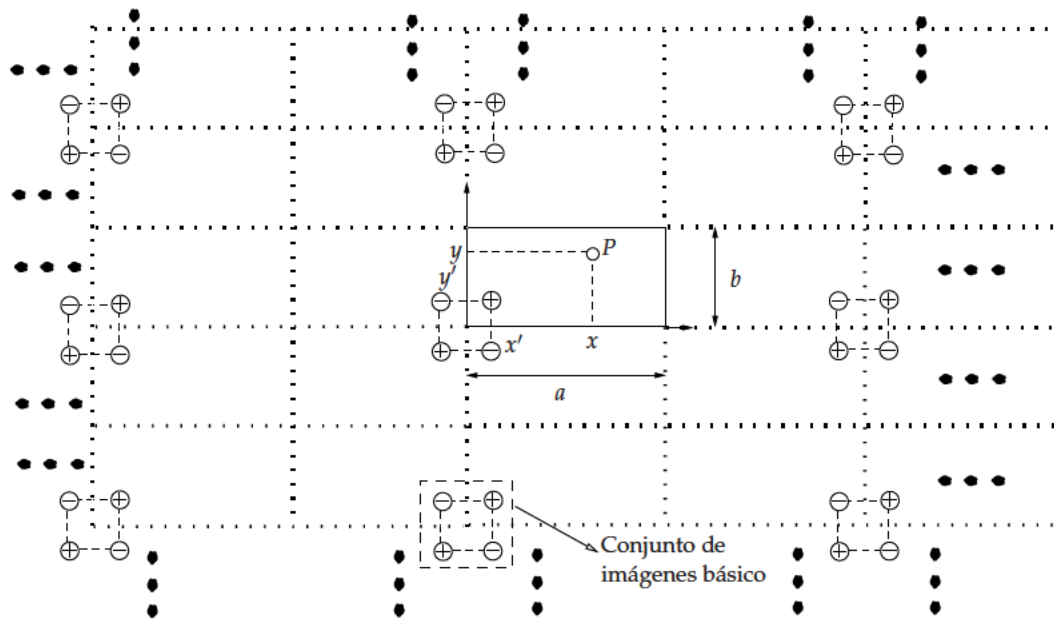


Figura 2.4: Serie de imágenes espaciales necesarias para satisfacer las condiciones de contorno en las paredes metálicas para una carga puntual en (x', y') .

Los diferentes conjuntos de imágenes básicas que se muestran en la Figura 2.4 pueden ser agrupados en capas de influencia alrededor de la carga original y las tres primeras imágenes. Esta agrupación en capas de influencia es apropiada para estudiar la convergencia de los métodos numéricos de suma de las series de imágenes. Además, el concepto de capas de conjuntos de imágenes básicas será empleado en el Capítulo 3 para mostrar un método de eliminación de la singularidad presente en las funciones de Green. De esta manera, los índices (m, n) son combinados para generar un único índice de capa k . Así, la capa $k = 1$ está compuesta por las cuatro primeras cargas (original y tres imágenes), la capa $k = 2$ contiene los siguientes ocho conjuntos de imágenes básicas alrededor del primero, la capa $k = 3$ los dieciséis siguientes, y así sucesivamente. De este modo, podemos observar que la Figura 2.4 contiene la primera y segundas capas de influencia. En la Figura 2.5 se observa la disposición de los conjuntos de imágenes básicas (puntos negros) en las diferentes capas. La representación de los índices (m, n) por un único índice k permite representar no solo la serie de imágenes espaciales, sino también los modos de la serie espectral. En este caso hay que emplear únicamente el primer cuadrante, debido a que el sumatorio empieza en $m = 0$ y $n = 0$, tal y como se aprecia en las ecuaciones (2.1, 2.2, 2.3).

La serie de imágenes espaciales converge de forma rápida especialmente cuando la frecuencia es baja. Cuando la frecuencia aumenta, comienza la excitación de ondas de superficie, que degradan la convergencia de la serie de imágenes cerca de la fuente. Por ello, el método de la suma de Poisson se va a aplicar únicamente a la parte cuasi-estática de la serie espectral de las ecuaciones (2.1, 2.2, 2.3). La parte cuasi-estática o asintótica no soporta ondas de superficie, de forma que no aparecerán problemas de convergencia al aplicar la suma de imágenes. La parte dinámica de la serie espectral se obtiene como resultado de restar la parte cuasi-estática a la serie completa. La parte dinámica converge rá-

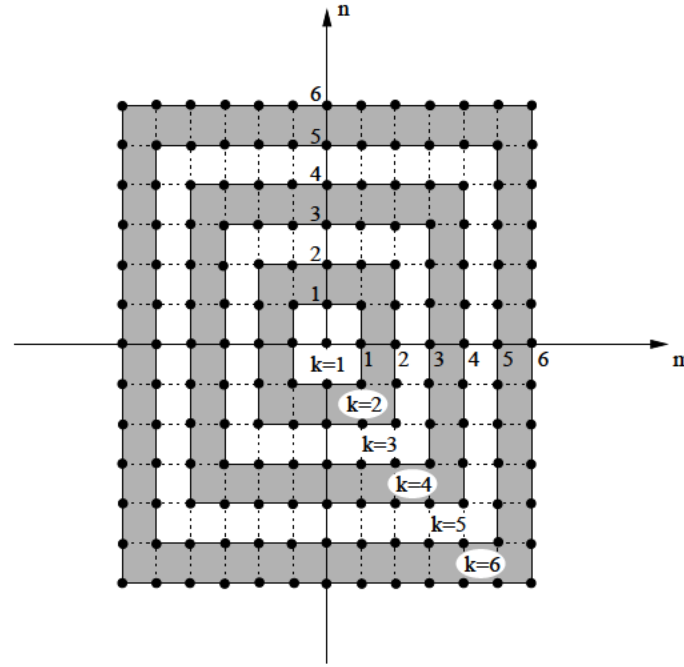


Figura 2.5: Representación del conjunto de capas de influencia.

pidamente de modo que se puede sumar de forma directa en el dominio espectral. El procedimiento de división de la serie espectral en dos partes, dinámica y estática, se denomina método de Kummer. En nuestro caso, la parte dinámica se sumará en el dominio espectral, mientras que la parte cuasi-estática será transformada al dominio espacial. A continuación detallamos el procedimiento llevado a cabo en cada una de las funciones de Green de los potenciales del medio encapsulado.

Si sustituimos la ecuación (2.5) en la ecuación (2.1) obtenemos la siguiente expresión de la función de Green del potencial eléctrico escalar:

$$G_{cV} = j\omega \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \sin(k_{x_m} x) \sin(k_{x_m} x') \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.24)$$

donde $V_{m,n}(z, z')$ es el voltaje calculado a partir del modelo circuital equivalente de la Figura 2.2 y $N_{b_{m,n}}$ es el factor de normalización asociado a las funciones modales. La parte cuasi-estática de 2.24 se puede extraer del voltaje asociado a la función de Green de forma que obtenemos:

$$G_{cV_0} = j\omega \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} V_{0,m,n} N_{b_{m,n}} \sin(k_{x_m} x) \sin(k_{x_m} x') \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.25)$$

Si a la serie completa de 2.24 se le resta la parte cuasi-estática anterior obtenemos la parte dinámica de la función de Green:

$$G_{cV_D} = j\omega \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} [V_{m,n} - V_{0,m,n}] N_{b_{m,n}} \sin(k_{x_m} x) \sin(k_{x_m} x') \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.26)$$

La serie modal completa se expresa como la suma de las dos partes, dinámica y cuasi-estática:

$$G_{cV} = G_{cV_D} + G_{cV_0} \quad (2.27)$$

Para calcular el voltaje correspondiente a la parte cuasi-estática $V_{0,m,n}$ se recurre al modelo circuital equivalente. En este caso se debe tener en cuenta que esta parte de la función de Green corresponde a un modo que se encuentra totalmente al corte. Debido a este hecho el modo no se puede propagar a lo largo de la estructura multicapa, y por tanto tampoco a lo largo de las líneas de transmisión que componen el modelo circuital. La onda electromagnética generada por la excitación se atenúa antes de alcanzar la primera interfaz dieléctrica. De este modo, el circuito general dibujado en las Figuras 2.1 y 2.2 queda reducido a dos líneas de transmisión de longitud infinita conectadas al generador de corriente I_g , tal como se observa en la Figura 2.6 a:

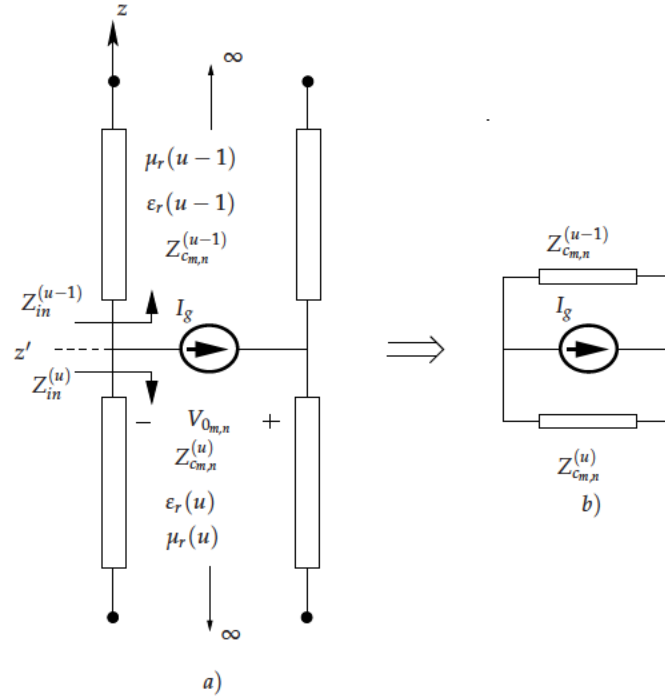


Figura 2.6: Modelo circuital equivalente necesario para calcular el voltaje de la parte cuasi-estática de las funciones de Green de los potenciales.

Debido a que las dos líneas de transmisión poseen longitud infinita, la impedancia de entrada de cada una de ellas es igual a la impedancia característica de la línea en cuestión. Así, el circuito equivalente se transforma en el circuito b) de la Figura (2.6), que permite calcular de forma sencilla el voltaje deseado:

$$Z_{in}^{(u)} = Z_{c_{m,n}}^{(u)} \quad (2.28a)$$

$$Z_{in}^{(u-1)} = Z_{c_{m,n}}^{(u-1)} \quad (2.28b)$$

$$V_{0,m,n} = \frac{1}{2\pi} \frac{Z_{c_{m,n}}^{(u)} Z_{c_{m,n}}^{(u-1)}}{Z_{c_{m,n}}^{(u)} + Z_{c_{m,n}}^{(u-1)}} \quad (2.29)$$

donde se ha considerado que la excitación I_g toma un valor igual a $1/2\pi$. La impedancia característica del medio depende del tipo de modo TE o TM considerado según se observa en la ecuación (2.10).

Una primera aproximación del voltaje de la parte cuasi-estática se consigue si consideramos que $m \rightarrow \infty$ y $n \rightarrow \infty$ en la expresión de la constante de propagación $\beta_{m,n}(s)$ de la ecuación (2.10) para una capa genérica s . Así obtenemos:

$$\beta_{m,n}^{(s)} = -jk_{\rho_{m,n}} \quad (2.30)$$

Sustituyendo las ecuaciones (2.30) y (2.10) en la ecuación (2.29) obtenemos los siguientes voltajes de los modos TE y TM:

$$V_{0,m,n}^{TE} = \frac{1}{2\pi} \frac{j d_{\omega}^{TE}}{k_{\rho_{m,n}}} \quad V_{0,m,n}^{TM} = \frac{1}{2\pi} \frac{k_{\rho_{m,n}}}{j d_{\omega}^{TM}} \quad (2.31a)$$

$$d_{\omega}^{TE} = \omega \mu_0 M(s) \quad d_{\omega}^{TM} = \omega \epsilon_0 E(s) \quad (2.31b)$$

$$M(s) = \frac{\mu_r(s) \mu_r(s-1)}{\mu_r(s) + \mu_r(s-1)} \quad E(s) = \epsilon_r(s-1) + \epsilon(s) \quad (2.31c)$$

La función de Green espectral de la parte estática \tilde{G}_{V_0} estará formada por una contribución debida al modo TE y otra debida al modo TM. Cada una de las contribuciones será fruto de la multiplicación del voltaje correspondiente ($V_{0,m,n}^{TM}$ o $V_{0,m,n}^{TE}$) por el factor de normalización $N_{b_{m,n}}$. La expresión final de la función de Green espectral deseada es:

$$\tilde{G}_{V_0} = \frac{\omega}{2\pi} \left[\frac{1}{d_{\omega}^{TM} k_{\rho}} + \frac{d_{\omega}^{TE}}{k_{\rho}^3} \right] \quad (2.32)$$

Una vez extraída la parte cuasi-estática de la serie espectral total, la serie resultante correspondiente a la parte dinámica converge rápidamente. En efecto, la parte dinámica del modo TE depende del término $1/k_{\rho_{m,n}}^5$ mientras que la debida al modo TM depende del término $1/k_{\rho_{m,n}}$. Para mejorar la convergencia de la parte dinámica del modo TM se realiza la extracción de un término de segundo orden de la parte cuasi-estática para el mencionado tipo de modo. A fin de conseguir la extracción

del nuevo término se recurre a la expansión de Taylor de la expresión de la constante de propagación descrita en la ecuación (2.10). Como se demuestra en [Álvarez Melcón, 1998] la extracción del nuevo término provoca que el voltaje del modo TM de la parte cuasi-estática tome el valor:

$$V_{0m,n}^{TM} = \frac{1}{j2\pi d_{\omega}^{TM}} \left[k_{\rho_{m,n}} + \frac{d_{\omega}^{TM} \hat{d}_{\omega}^{TM}}{2k_{\rho_{m,n}}} \right] \quad (2.33a)$$

$$\hat{d}_{\omega}^{TM} = \omega \mu_0 \frac{\hat{E}(s)}{E(s)} \quad (2.33b)$$

$$\hat{E}(s) = -\frac{\varepsilon_r^2(s-1)\mu_r(s-1) + \varepsilon_r^2(s)\mu_r(s)}{\varepsilon_r(s-1) + \varepsilon_r(s)} \quad (2.33c)$$

donde d_{ω}^{TM} y $\hat{E}(s)$ son valores definidos en (2.31) y \hat{d}_{ω}^{TM} es un término obtenido a partir del desarrollo de Taylor. Como se observa en la ecuación anterior ahora se añade a la parte cuasi-estática un segundo término que depende de $1/k_{\rho_{m,n}}^3$. La serie dinámica del modo TM resultado de sustraer la parte cuasi-estática calculada en (2.33) converge como $1/k_{\rho_{m,n}}^5$. La función de Green espectral correspondiente a la parte cuasi-estática es finalmente:

$$\tilde{G}_{V_0} = \frac{\omega}{2\pi} \left[\frac{1}{d_{\omega}^{TM} k_{\rho}} + \frac{d_{\omega}^{TE}}{k_{\rho}^3} + \frac{\hat{d}_{\omega}^{TM}}{2k_{\rho_{m,n}}^3} \right] \quad (2.34)$$

Una vez obtenida la función de Green en el dominio espectral podemos obtener, a partir de su anti-transformación, la función de Green espacial de la parte cuasi-estática. Para los modos TE,TM de índice (m,n) dicha función de Green corresponde a los campos electromagnéticos generados por una carga puntual desplazada en los ejes x e y la cantidad adecuada marcada por los índices mencionados. Como el sistema de cargas reemplaza a las paredes metálicas, la estructura multicapa poseerá dimensiones transversales infinitas. Por tanto, para calcular la función de Green espacial básica necesaria para generar la serie de imágenes se debe emplear la integral de Sommerfeld [Gradshteyn y Ryzhik, 1965]:

$$\begin{aligned} G_{V_0} &= \frac{\omega}{2\pi} \left[\int_0^{\infty} \frac{J_0(k_{\rho}\rho)}{d_{\omega}^{TM}} dk_{\rho} + \int_0^{\infty} J_0(k_{\rho}\rho) \frac{d_{\omega}^{TE}}{k_{\rho}^2} dk_{\rho} + \int_0^{\infty} J_0(k_{\rho}\rho) \frac{\hat{d}_{\omega}^{TM}}{2k_{\rho}^2} dk_{\rho} \right] \\ &= \frac{\omega}{2\pi} \left[\frac{1}{d_{\omega}^{TM} \rho} - \left(d_{\omega}^{TE} + \frac{\hat{d}_{\omega}^{TM}}{2} \right) \rho \right] \end{aligned} \quad (2.35)$$

Finalmente, se puede emplear la función de Green espacial de la ecuación (2.35) para transformar la serie espectral de la parte cuasi-estática mostrada en la ecuación (2.25) en una serie de imágenes de rápida convergencia cerca de la fuente. La ecuación final que expresa el diferente procesamiento de la parte dinámica (suma espectral) y cuasi-estática (suma espacial) se muestra a continuación:

$$\begin{aligned}
G_{cV} = & j\omega \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} [V_{m,n} - V_{0,m,n}] N_{b_{m,n}} \sin(k_{x_m} x) \sin(k_{x_m} x') \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (\text{Parte Dinámica}) \\
& + \frac{ab}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [G_{V_0}(x + x' + 2ma, y + y' + 2nb) - G_{V_0}(x - x' + 2ma, y + y' + 2nb) \\
& - G_{V_0}(x + x' + 2ma, y - y' + 2nb) + G_{V_0}(x - x' + 2ma, y - y' + 2nb)] \quad (\text{Parte Cuasi-estática})
\end{aligned} \tag{2.36}$$

$$\tilde{G}_{A_0}^{xx} = \tilde{G}_{A_0}^{yy} = \frac{1}{2\pi\omega} \frac{d_{\omega}^{TE}}{k_{\rho}} \quad (2.39)$$

La función de Green de espacio libre definida en el dominio espacial se calcula a partir de la expresión anterior mediante la integral de Sommerfeld:

$$G_{A_0}^{xx} = G_{A_0}^{yy} = \frac{d_{\omega}^{TE}}{2\pi\omega} \int_0^{\infty} J_0(k_{\rho}\rho) dk_{\rho} = \frac{1}{2\pi\omega} \frac{d_{\omega}^{TE}}{\rho} \quad (2.40)$$

Ahora ya se pueden representar las funciones de Green del potencial vector magnético como suma de la parte dinámica en el dominio espectral más la serie periódica de las funciones de Green espaciales de la parte estática:

$$\begin{aligned} G_{cA}^{xx} = & + \frac{1}{j\omega} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} [V_{m,n}^{TE} - V_{0,m,n}^{TE}] N_{a_{m,n}}^{TE} \cos(k_{x_m}x) \cos(k_{x_m}x') \sin(k_{y_n}y) \sin(k_{y_n}y') \\ & + \frac{ab}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [G_{A_0}^{xx}(x+x'+2ma, y+y'+2nb) + G_{A_0}^{xx}(x-x'+2ma, y+y'+2nb) \\ & - G_{A_0}^{xx}(x+x'+2ma, y-y'+2nb) - G_{A_0}^{xx}(x-x'+2ma, y-y'+2nb)] \end{aligned} \quad (2.41a)$$

$$\begin{aligned} G_{cA}^{yy} = & - \frac{1}{j\omega} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} [V_{m,n}^{TE} - V_{0,m,n}^{TE}] N_{a_{m,n}}^{TE} \sin(k_{x_m}x) \sin(k_{x_m}x') \cos(k_{y_n}y) \cos(k_{y_n}y') \\ & + \frac{ab}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [G_{A_0}^{yy}(x+x'+2ma, y+y'+2nb) - G_{A_0}^{yy}(x-x'+2ma, y+y'+2nb) \\ & + G_{A_0}^{yy}(x+x'+2ma, y-y'+2nb) - G_{A_0}^{yy}(x-x'+2ma, y-y'+2nb)] \end{aligned} \quad (2.41b)$$

La única diferencia con respecto al procedimiento aplicado a G_{cV} estriba en que la extracción de un término de segundo orden provoca problemas de convergencia [Álvarez Melcón, 1998]. Por tanto, en la funciones de Green de los potenciales vector magnético únicamente se extraerá un término de la parte cuasi-estática. De este modo la serie dinámica modal poseerá una convergencia de tipo $1/k_{\rho_{m,n}}^3$.

A continuación se muestra un breve estudio cualitativo sobre la convergencia de los métodos numéricos descritos. Para ello tomamos una estructura multicapa apantallada como la dibujada en la Figura 2.7. El número de capas de influencia es empleado para medir el grado de convergencia del método de suma de imágenes espaciales. Cuanto mayor es el número de capas de influencia necesarias para alcanzar una cierta precisión, mayor es el número de grupos básicos, y como consecuencia más lenta será la convergencia en la suma de imágenes. Se han estudiado dos situaciones diferentes. En el primer caso el dipolo se encuentra situado en el centro de la cavidad, lejos de las paredes metálicas. Las curvas de nivel de la Figura 2.8 muestran que el método de suma de imágenes necesita un número mínimo de capas de influencia cuando los puntos observación se ubican cerca de la fuente. Conforme los puntos observación se alejan del punto fuente, la convergencia empeora, aunque la degradación no es simétrica respecto al punto fuente debido a la distribución de las imágenes espaciales. En la segunda simulación el dipolo se encuentra localizado cerca de una esquina

de la caja de la Figura 2.7 y por tanto cerca de las paredes metálicas. En la Figura 2.9 se aprecia un deterioro de la convergencia en la suma de imágenes espaciales análogo al obtenido en el primer caso. Conforme los puntos observación se distancian del punto fuente aumenta el número de capas de influencia requeridas para alcanzar la precisión fijada. En ambos casos los puntos observación se distribuyen en una rejilla regular en todo el plano XY de la interfaz entre substratos.

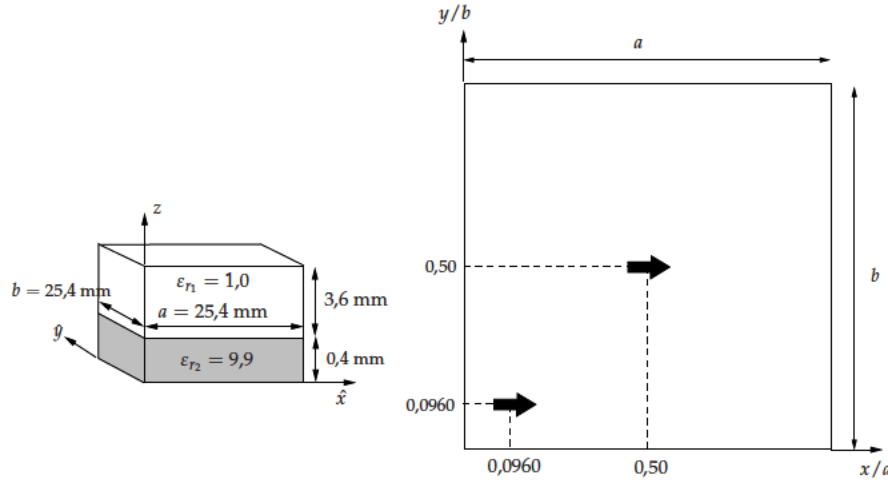


Figura 2.7: Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado en el centro del plano XY.

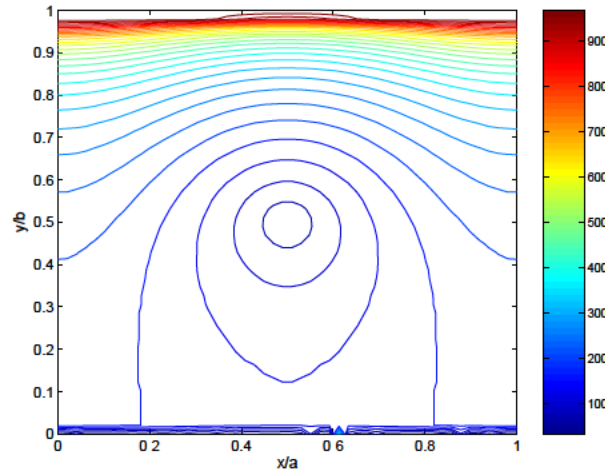


Figura 2.8: Curvas de nivel correspondientes al número de capas de influencia necesarias para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

2.4. Método de suma en el dominio espectral: suma por partes

La técnica descrita en la Sección 2.3 presenta problemas de convergencia cuando la distancia entre el punto observación y el punto fuente aumenta. Dichos problemas afectan especialmente a la técnica de suma de imágenes. Para poder completar el cálculo de las funciones de Green para todas distancias observador-fuente se aplicarán un conjunto de transformaciones matemáticas sobre

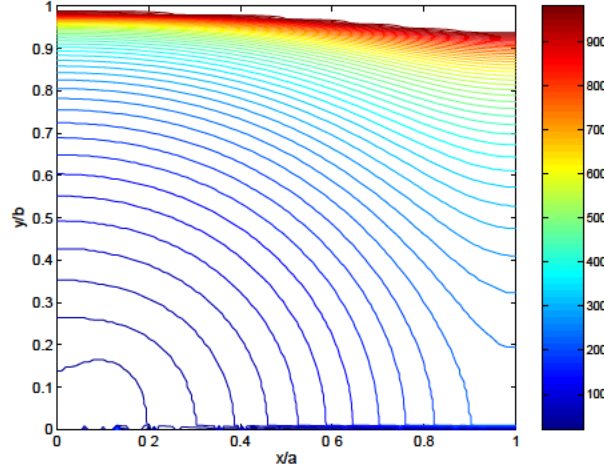


Figura 2.9: Curvas de nivel correspondientes al número de capas de influencia necesarias para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra cerca de una esquina de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

la serie espectral. El procedimiento permite la evaluación de la suma modal de forma eficaz cuando el punto observación se encuentre lejos del punto fuente. Además, en este método no será necesaria la separación en parte cuasi-estática y dinámica.

En un primer paso se va a aplicar la técnica de la suma por partes a una serie unidimensional genérica. Las expresiones calculadas para la serie unidimensional serán aplicadas a las series bidimensionales que forman las funciones de Green de los potenciales. De este modo definimos la serie infinita unidimensional como:

$$S_{\infty} = \sum_{n=0}^{\infty} \tilde{G}_n f_n \quad (2.42)$$

donde \tilde{G}_n representa una función de Green definida en el dominio espectral y f_n una función sinusoidal. En primer lugar dividiremos la serie espectral en dos términos, una suma parcial finita S_{N-1} y una suma infinita de términos R_N denominada resto, así obtenemos:

$$S_{\infty} = S_{N-1} + R_N \quad S_{N-1} = \sum_{n=0}^{N-1} \tilde{G}_n f_n \quad R_N = \sum_{n=N}^{\infty} \tilde{G}_n f_n \quad (2.43)$$

La suma parcial es evaluada directamente, mientras que el resto debe ser tratado numéricamente. La clave para acelerar la convergencia de la serie infinita que forma el resto consiste en expresar dicha serie en función de series que únicamente involucren a la función sinusoidal f_n . La suma del resto se desarrolla como:

$$R_N = \sum_{n=N}^{\infty} \tilde{G}_n f_n = \tilde{G}_N f_N + \tilde{G}_{N+1} f_{N+1} + \tilde{G}_{N+2} f_{N+2} + \tilde{G}_{N+3} f_{N+3} + \dots \quad (2.44)$$

La ecuación (2.44) puede a su vez ser re-escrita como:

$$\begin{aligned}
 R_N = & \tilde{G}_N (f_N + f_{N+1} + f_{N+2} + f_{N+3} + \dots) - \tilde{G}_N (f_{N+1} + f_{N+2} + f_{N+3} + \dots) \\
 & + \tilde{G}_{N+1} (f_{N+1} + f_{N+2} + f_{N+3} + \dots) - \tilde{G}_{N+1} (f_{N+2} + f_{N+3} + \dots) \\
 & + \tilde{G}_{N+2} (f_{N+2} + f_{N+3} + \dots) - \tilde{G}_{N+2} (f_{N+3} + \dots) + \dots
 \end{aligned} \quad (2.45)$$

La ecuación anterior mantiene el valor del resto R_N ya que los nuevos términos introducidos son sumados y restados a la serie original. La inclusión de los nuevos términos permite escribir la serie del siguiente modo:

$$\begin{aligned}
 R_N = & \tilde{G}_N \sum_{n=N}^{\infty} f_n - \tilde{G}_N \sum_{n=N+1}^{\infty} f_n + \tilde{G}_{N+1} \sum_{n=N+1}^{\infty} f_n \\
 & - \tilde{G}_{N+1} \sum_{n=N+2}^{\infty} f_n + \tilde{G}_{N+2} \sum_{n=N+2}^{\infty} f_n - \tilde{G}_{N+2} \sum_{n=N+3}^{\infty} f_n + \dots
 \end{aligned} \quad (2.46)$$

Los términos que componen la serie anterior se pueden re-agrupar de forma que aparezcan series infinitas de la función sinusoidal f_n

$$R_N = \tilde{G}_N \sum_{n=N}^{\infty} f_n + (\tilde{G}_{N+1} - \tilde{G}_N) \sum_{n=N+1}^{\infty} f_n + (\tilde{G}_{N+2} - \tilde{G}_{N+1}) \sum_{n=N+2}^{\infty} f_n + \dots \quad (2.47)$$

Finalmente tras manipular la ecuación anterior llegamos a la expresión:

$$R_N = \tilde{G}_N \sum_{n=N}^{\infty} f_n + \sum_{n=N}^{\infty} (\tilde{G}_{n+1} - \tilde{G}_n) \sum_{k=n+1}^{\infty} f_k \quad (2.48)$$

La ecuación (2.48) nos indica que el resto R_N se puede expresar por medio de series auxiliares formadas por la función sinusoidal f_n y por términos que involucran diferencias de la función \tilde{G}_n de lenta variación. Se puede definir la siguiente notación:

$$\tilde{G}_n^{(-1)} = \tilde{G}_n \quad \tilde{G}_n^{(-2)} = \tilde{G}_{n+1}^{(-1)} - \tilde{G}_n^{(-1)} \quad (2.49a)$$

$$f_n^{(+1)} = f_n \quad f_n^{(+2)} = \sum_{k=n+1}^{\infty} f_k^{(+1)} \quad (2.49b)$$

Si se aplica la anterior notación a la ecuación de definición del resto R_N :

$$\begin{aligned}
R_N &= \sum_{n=N}^{\infty} \tilde{G}_n^{(-1)} f_n^{(+1)} \\
&= \tilde{G}_N^{(-1)} f_{N-1}^{(+2)} + \sum_{n=N}^{\infty} \tilde{G}_n^{(-2)} f_n^{(+2)}
\end{aligned} \tag{2.50}$$

El procedimiento puede volver a aplicarse al segundo término de la ecuación anterior:

$$\begin{aligned}
R_N &= \tilde{G}_N^{(-1)} f_{N-1}^{(+2)} + \tilde{G}_N^{(-2)} \sum_{n=N}^{\infty} f_n^{(+2)} + \sum_{n=N}^{\infty} [\tilde{G}_{n+1}^{(-2)} - \tilde{G}_n^{(-2)}] \sum_{k=n+1}^{\infty} f_k^{(+2)} \\
&= \tilde{G}_N^{(-1)} f_{N-1}^{(+2)} + \tilde{G}_N^{(-2)} f_n^{(+3)} + \sum_{n=N}^{\infty} \tilde{G}_n^{(-3)} f_n^{(+3)}
\end{aligned} \tag{2.51}$$

El proceso mostrado en la ecuación inmediatamente anterior se puede repetir indefinidamente para obtener la siguiente expresión del resto R_N :

$$R_N = \sum_{n=N}^{\infty} \tilde{G}_n^{(-1)} f_n^{(+1)} = \sum_{i=1}^{\infty} \tilde{G}_N^{(-i)} f_{N-1}^{(i+1)} \tag{2.52}$$

en la que se han utilizado las siguientes funciones fruto de la generalización de las ecuaciones mostradas en (2.49):

$$\tilde{G}_n^{(-1)} = \tilde{G}_n \quad \tilde{G}_n^{(-i)} = \tilde{G}_{n+1}^{(-i+1)} - \tilde{G}_n^{(-i+1)} \quad \text{Para todo } i = 2, 3, 4, \dots \tag{2.53a}$$

$$f_n^{(+1)} = f_n \quad f_n^{(+i)} = \sum_{k=n+1}^{\infty} f_k^{(i-1)} \quad \text{Para todo } i = 2, 3, 4, \dots \tag{2.53b}$$

La suma de un número P reducido de términos $\tilde{G}_N^{(-i)} f_{N-1}^{(i+1)}$ permitirá evaluar el resto R_N con cierta precisión:

$$R_N \approx \sum_{i=1}^P \tilde{G}_N^{(-i)} f_{N-1}^{(i+1)} \tag{2.54}$$

Además, la anterior expresión está compuesta por diferencias parciales de la función $\tilde{G}_N^{(-i)}$ y de sumas infinitas de la función sinusoidal $f_{N-1}^{(i+1)}$. En el Apéndice A se demuestra que dichas sumas infinitas pueden calcularse de forma analítica, de modo que el número total de operaciones necesario para calcular la serie completa se verá reducido drásticamente.

La técnica descrita previamente puede ser aplicada a las series bidimensionales de las funciones de Green de las ecuaciones (2.1,2.2,2.3). Siguiendo la metodología empleada con la serie unidimensional expresamos el resto como:

$$R_{M,N} = \sum_{m=M}^{\infty} \sum_{n=N}^{\infty} \tilde{G}_{m,n} f_m g_n \quad (2.55)$$

donde $G_{m,n}$ es la correspondiente función de Green del dominio espectral, y f_m y g_n son funciones consistentes en productos de funciones sinusoidales. Asimismo, empleamos una notación similar a la utilizada en el caso unidimensional:

$$\tilde{G}_{m,n}^{(-1,-1)} = \tilde{G}_{m,n}; \quad f_m^{(+1)} = f_m; \quad g_n^{(+1)} = g_n \quad (2.56)$$

Así el resto toma la expresión:

$$R_{M,N} = \sum_{m=M}^{\infty} f_m^{(+1)} \sum_{n=N}^{\infty} \tilde{G}_{m,n}^{(-1,-1)} g_n^{(+1)} \quad (2.57)$$

A partir de la ecuación anterior podemos definir la suma parcial unidimensional:

$$\Psi(m, N) = \sum_{n=N}^{\infty} \tilde{G}_{m,n}^{(-1,-1)} g_n^{(+1)} \quad (2.58)$$

De esta forma el resto queda expresado como una serie unidimensional dependiente de $\Psi(m, N)$:

$$R_{M,N} = \sum_{m=M}^{\infty} \Psi(m, N) f_m^{(+1)} \quad (2.59)$$

Por tanto, el problema original consistente en el cálculo de una serie bidimensional equivale a la evaluación de dos series unidimensionales. Para emplear la técnica de la suma por partes en la primera de las series unidimensionales debemos definir las siguientes diferencias y sumas parciales:

$$\tilde{G}_{m,n}^{(-1,-k)} = \tilde{G}_{m,n+1}^{(-1,-k+1)} - \tilde{G}_{m,n}^{(-1,-k+1)}; \quad g_n^{(+k)} = \sum_{i=n+1}^{\infty} g_i^{(k-1)} \quad \text{Para todo: } k = 2, 3, 4, \dots \quad (2.60)$$

La suma unidimensional $\Psi(m, N)$ se calcula como:

$$\Psi(m, N) = \sum_{k=1}^{\infty} \tilde{G}_{m,N}^{(-1,-k)} g_{N-1}^{k+1} \quad (2.61)$$

Introduciendo la ecuación (2.61) en la ecuación (2.59) obtenemos:

$$R_{M,N} = \sum_{m=M}^{\infty} \sum_{k=1}^{\infty} \tilde{G}_{m,N}^{(-1,-k)} g_{k+1}^{(N-1)} f_m^{(+1)} \quad (2.62)$$

Reordenando los términos llegamos a una expresión en la que podemos volver a aplicar la técnica de suma por partes para series unidimensionales:

$$R_{M,N} = \sum_{k=1}^{\infty} g_{k+1}^{(N-1)} \sum_{m=M}^{\infty} \tilde{G}_{m,N}^{(-1,-k)} f_m^{(+1)} \quad (2.63)$$

De forma análoga a la primera serie unidimensional definimos los términos:

$$\tilde{G}_{m,n}^{(-i,-k)} = \tilde{G}_{m+1,n}^{(-i+1,-k)} - \tilde{G}_{m,n}^{(-i+1,-k)}; \quad f_n^{(+i)} = \sum_{k=n+1}^{\infty} f_k^{(i-1)} \quad \text{Para todo } \begin{cases} i = 1, 2, 3, \dots \\ k = 1, 2, 3, \dots \end{cases} \quad i = k \neq 1 \quad (2.64)$$

Una vez introducidos en la ecuación 2.61:

$$R_{M,N} = \sum_{i=1}^{\infty} \sum_{k=1}^{\infty} \tilde{G}_{M,N}^{(-i,-k)} f_{M-1}^{(i+1)} g_{N-1}^{(k+1)} \quad (2.65)$$

De este modo el cálculo de la serie original queda reducida a la evaluación de las funciones de diferencias $\tilde{G}_{m,n}^{(-i,-k)}$ y de las sumas infinitas de carácter analítico $f_m^{(+i)}$ y $g_n^{(+k)}$. Para aplicar la técnica de la suma por partes a las funciones de Green de los potenciales es más conveniente representar los productos de senos y cosenos por funciones de tipo coseno. Así, los términos de la función de Green del potencial eléctrico escalar se pueden reordenar como:

$$G_{cV} = j\omega \sum_{m=1}^{\infty} \sin(k_{x_m} x) \sin(k_{x_m} x') \sum_{n=1}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.66)$$

Al igual que el ejemplo genérico anterior definimos una serie unidimensional que llamamos $Y(m)$:

$$Y(m) = \sum_{n=1}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \sin(k_{y_n} y) \sin(k_{y_n} y') \quad (2.67)$$

Si introducimos la ecuación anterior en (2.66) obtenemos la siguiente expresión:

$$G_{cV} = j\omega \sum_{m=1}^{\infty} Y(m) \sin(k_{x_m} x) \sin(k_{x_m} x') \quad (2.68)$$

Aplicando las transformaciones trigonométricas apropiadas para representar las series en función de cosenos se obtiene:

$$Y(m) = \frac{j\omega}{2} \left\{ \sum_{n=1}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \cos[k_{y_n}(y - y')] - \sum_{n=1}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \cos[k_{y_n}(y + y')] \right\} \quad (2.69a)$$

$$G_{cV} = \frac{1}{2} \left\{ \sum_{m=1}^{\infty} Y(m) \cos[k_{x_m}(x - x')] - \sum_{m=1}^{\infty} Y(m) \cos[k_{x_m}(x + x')] \right\} \quad (2.69b)$$

A partir de la función Y se consiguen funciones de resto análogas a la mostrada en (2.58):

$$\Psi_1(m, N) = \frac{j\omega}{2} \sum_{n=N}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \cos[k_{y_n}(y - y')] \quad (2.70a)$$

$$\Psi_2(m, N) = \frac{j\omega}{2} \sum_{n=N}^{\infty} V_{m,n}(z, z') N_{b_{m,n}} \cos[k_{y_n}(y + y')] \quad (2.70b)$$

La introducción de las dos funciones anteriores en la ecuación (2.69b) proporciona la descomposición de la función de Green del potencial eléctrico escalar en cuatro series tipo coseno con la forma expresada en la ecuación (2.59). Cada una de estas series bidimensionales es evaluada mediante la técnica de suma por partes que implica el cómputo de una suma parcial y un resto (ver ecuación (2.65)). Como ya se ha comentado dicho resto es calculado rápidamente mediante la técnica de suma por partes gracias al carácter analítico de las sumas infinitas sucesivas de las funciones sinusoidales f_n y g_n (ver Apéndice A).

Un procedimiento similar se emplea con las funciones de Green del potencial vector magnético G_{cA}^{xx} y G_{cA}^{yy} . En primer lugar, podemos reescribir las ecuaciones (2.2) y (2.3):

$$G_{cA}^{xx} = \frac{1}{j\omega} \sum_{m=0}^{\infty} \cos(k_{x_m}x) \cos(k_{x_m}x') \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \sin(k_{y_n}y) \sin(k_{y_n}y') \quad (2.71a)$$

$$G_{cA}^{yy} = -\frac{1}{j\omega} \sum_{m=0}^{\infty} \sin(k_{x_m}x) \sin(k_{x_m}x') \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos(k_{y_n}y) \cos(k_{y_n}y') \quad (2.71b)$$

Al igual que en el caso del potencial eléctrico escalar definimos las siguientes series unidimensionales:

$$Y^{xx}(m) = \frac{1}{j\omega} \left\{ \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \sin(k_{y_n}y) \sin(k_{y_n}y') \right\} \quad (2.72a)$$

$$Y^{yy}(m) = -\frac{1}{j\omega} \left\{ \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos(k_{y_n}y) \cos(k_{y_n}y') \right\} \quad (2.72b)$$

Mediante relaciones trigonométricas, las anteriores funciones (Y) se pueden describir como funciones tipo coseno:

$$Y^{xx}(m) = \frac{1}{2j\omega} \left\{ \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos [k_{y_n}(y - y')] - \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos [k_{y_n}(y + y')] \right\} \quad (2.73a)$$

$$Y^{yy}(m) = -\frac{1}{2j\omega} \left\{ \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos [k_{y_n}(y - y')] + \sum_{n=0}^{\infty} V_{m,n}(z, z') N_{a_{m,n}} \cos [k_{y_n}(y + y')] \right\} \quad (2.73b)$$

De esta forma las funciones de Green deseadas se pueden expresar totalmente en función de funciones tipo coseno:

$$G_{cA}^{xx} = \left\{ \sum_{m=0}^{\infty} Y^{xx}(m) \cos [k_{x_m}(x - x')] + \sum_{m=0}^{\infty} Y^{xx}(m) \cos [k_{x_m}(x + x')] \right\} \quad (2.74a)$$

$$G_{cA}^{yy} = \left\{ \sum_{m=0}^{\infty} Y^{yy}(m) \cos [k_{x_m}(x - x')] - \sum_{m=0}^{\infty} Y^{yy}(m) \cos [k_{x_m}(x + x')] \right\} \quad (2.74b)$$

De forma análoga a la función de Green del potencial eléctrico escalar, se pueden definir las correspondientes funciones resto de G_{cA}^{xx} y G_{cA}^{yy} que permiten evaluar de forma eficiente la serie completa. Debido a que las ecuaciones (2.74) dependen de funciones tipo coseno, las secuencias sucesivas se calculan analíticamente gracias a la ecuación (A.5) del Apéndice A.

Al igual que en el método de las imágenes, mostraremos la convergencia del método de suma por partes para la estructura apantallada multicapa particular dibujada en la Figura 2.7. Asimismo tomaremos dos dipolos diferentes, uno situado cerca de las paredes metálicas y otro situado en el centro de la estructura, por tanto alejado de las paredes laterales. El parámetro para estudiar la convergencia del método es el orden doble (M, N) del resto $R_{M,N}$. Debido a que el sumatorio interno y externo de dicho resto toman el mismo valor ($M = N$), el parámetro de estudio toma un único valor para cada par de puntos observación y fuente. El orden del resto indica el grado de rapidez en la convergencia hacia la solución. Un orden mayor conlleva la adición de más términos en la suma parcial que completa la serie junto con el resto $R_{M,N}$, de forma que la precisión se alcanza más lentamente. En las Figuras 2.10 y 2.11 se observa que el método de suma por partes converge rápidamente cuando el punto observación se encuentra lejos del punto fuente. En cambio, conforme el punto observación se acerca al punto fuente, es necesario un orden mayor para alcanzar la precisión requerida. Los puntos observación se distribuyen en una rejilla regular a lo largo del plano XY de la cavidad.

Las Figuras 2.8 y 2.9 muestran que el algoritmo de suma de imágenes para la parte cuasi-estática converge rápidamente cuando el punto observación se sitúa en las cercanías del punto fuente. En cambio, el método de suma por partes converge correctamente cuando el punto observación se ubica lejos del punto fuente, tal como se aprecia en las Figuras 2.10 y 2.11. Por tanto, se define un umbral en torno del punto fuente para conmutar de un método a otro, de modo que la evaluación de las

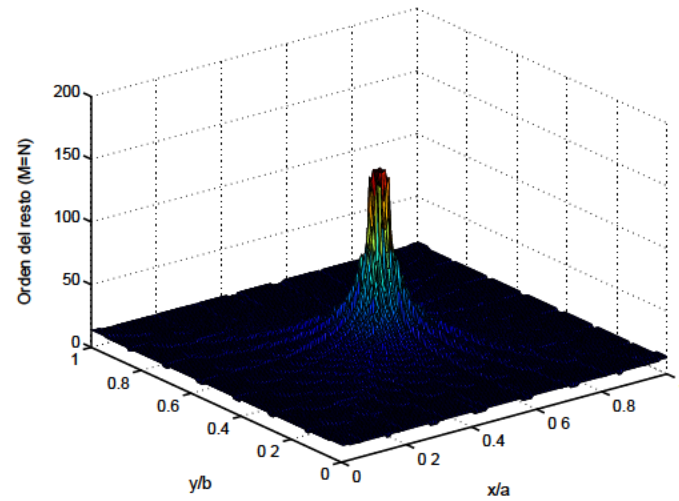


Figura 2.10: Orden del resto ($M=N$) necesario para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

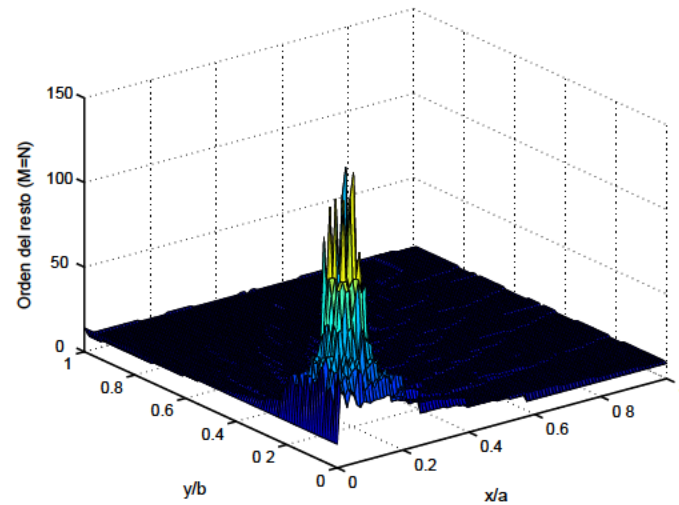


Figura 2.11: Orden del resto ($M=N$) necesario para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra cerca de una esquina de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

funciones de Green sea lo más eficiente posible. Así, el umbral delimita dos zonas, la zona cercana a la fuente que denominaremos de máxima singularidad (MS) y la zona alejada de la fuente o zona de débil singularidad (DS). Si el punto observación pertenece a la zona MS se empleará el método híbrido, si por el contrario el punto observación se localiza en la zona DS se empleará el algoritmo de suma por partes.

Las Figuras 2.12 y 2.13 muestran la incidencia del valor de umbral en el tiempo empleado para calcular las funciones de Green. Concretamente, la Figura 2.12 relaciona el valor de umbral con el

tiempo de cálculo a lo largo de cien puntos observación situados en la recta $y/b = 0,50$ de la cavidad de la Figura 2.7 cuando el punto fuente se encuentra en el centro de la cavidad. En la Figura 2.13 los puntos observación pertenecen a la recta $y/b = 0,0960$ y el punto fuente es el situado cerca de las paredes tal como se aprecia en la Figura 2.7.

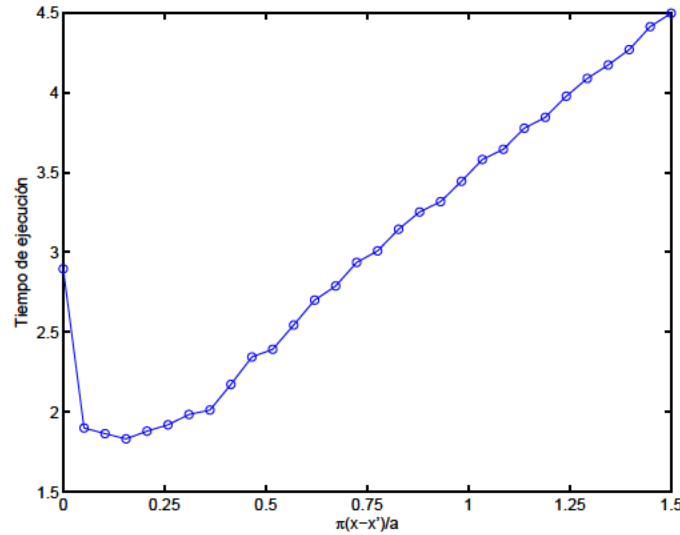


Figura 2.12: Tiempo de ejecución empleado para calcular las funciones de Green en función del umbral U_x cuando el punto fuente se encuentra situado en el centro de la cavidad de la Figura 2.7. Los puntos observación pertenecen a la recta $y/b = 0,50$ situada sobre el punto fuente. La frecuencia es 10 GHz.

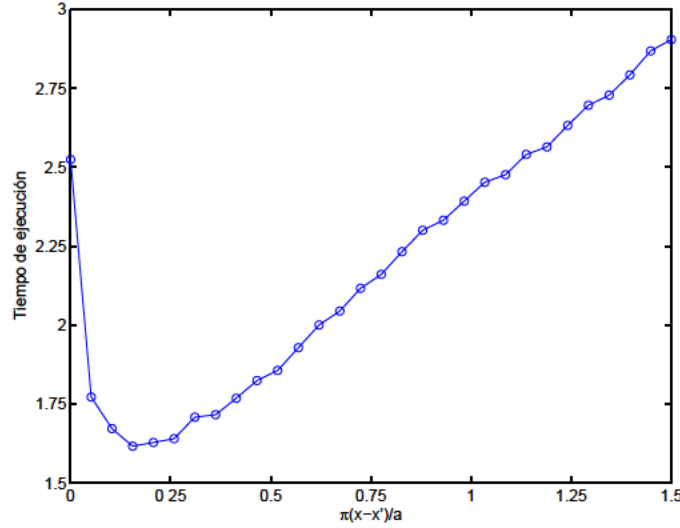


Figura 2.13: Tiempo de ejecución empleado para calcular las funciones de Green en función del umbral U_x cuando el punto fuente se encuentra situado cerca de las paredes de la cavidad de la Figura 2.7. Los puntos observación pertenecen a la recta $y/b = 0,0960$ situada sobre el punto fuente. La frecuencia es 10 GHz.

Se observa que si el umbral posee un valor pequeño entonces el método de suma por partes es utilizado cerca de la fuente donde resulta ineficiente. Si por el contrario el umbral toma un valor tal que el método híbrido es empleado lejos de la fuente se produce un aumento en el tiempo de cálculo.

Este efecto es debido a que la suma de imágenes espaciales converge lentamente en puntos alejados de la fuente. El estudio de rectas de abscisa constante conduce a gráficas similares a las mostradas y por tanto a las mismas conclusiones. Por lo tanto el umbral en el eje X (U_x) y el eje Y (U_y) deberá situarse aproximadamente entre los siguientes límites para alcanzar una mejor eficiencia en el cálculo de las funciones de Green:

$$0,05 < U_x < 0,25 \quad 0,05 < U_y < 0,25 \quad (2.75)$$

Ambos umbrales tomarán siempre el mismo valor, $U_x = U_y$. Una vez fijados los umbrales, cada punto observación deberá ser asignado a la zona MS o a la zona DS según el resultado de la siguiente expresión:

$$\begin{aligned} (x, y) \in \text{zona MS si } \left| \frac{\pi(x - x')}{a} \right| < U_x \text{ y } \left| \frac{\pi(y - y')}{b} \right| < U_y \\ (x, y) \in \text{zona DS en caso contrario} \end{aligned} \quad (2.76)$$

2.5. Propiedades de las funciones de Green

Ciertas propiedades de las funciones de Green influyen decisivamente en su aproximación por medio de redes neuronales. Algunas de ellas, como la presencia de un comportamiento singular alrededor de la fuente, pueden llegar a impedir el correcto entrenamiento de las redes neuronales. Otras, como la existencia de simetrías en la cavidad, permiten reducir el número de puntos de los conjuntos de entrenamiento y validación, de manera que el proceso general de aproximación se acelera de forma notable.

2.5.1. Singularidad en la fuente

A pesar de que las funciones de Green de los campos electromagnéticos exhiben singularidades más acentuadas que las funciones de Green de los potenciales, éstas siguen presentando un comportamiento abrupto en las cercanías de la fuente. Así, en las Figuras 2.14 y 2.15 se observa dicho comportamiento en la función de Green del potencial eléctrico escalar (G_{cV}) para dos posiciones diferentes del punto fuente. Los puntos observación se distribuyen en una rejilla regular que llena la interfaz entre las dos capas dieléctricas de la cavidad de la Figura 2.7.

El rápido crecimiento de las funciones de Green cerca de la fuente dificulta su aproximación por parte de las redes neuronales, ya que empeora de forma decisiva su capacidad de generalización. Las redes neuronales son capaces de generar nuevos datos de salida a partir de la interpolación de los valores de las funciones objetivo en los puntos de entrenamiento. El número de datos de entrenamiento debe ser tal que permita la representación fiel del comportamiento de las funciones objetivo en todo el espacio. De este modo, ante un nuevo dato de entrada la red neuronal calculará un valor de salida adecuado, gracias a que posee suficiente información sobre las funciones deseadas alrededor de ese

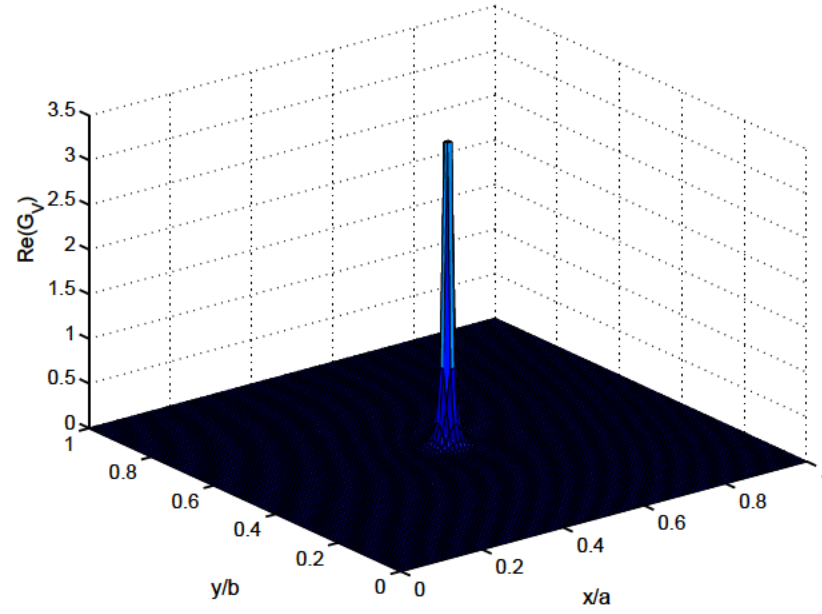


Figura 2.14: Parte real de la función de Green del potencial eléctrico escalar G_{cV} correspondiente a un dipolo excitación situado en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

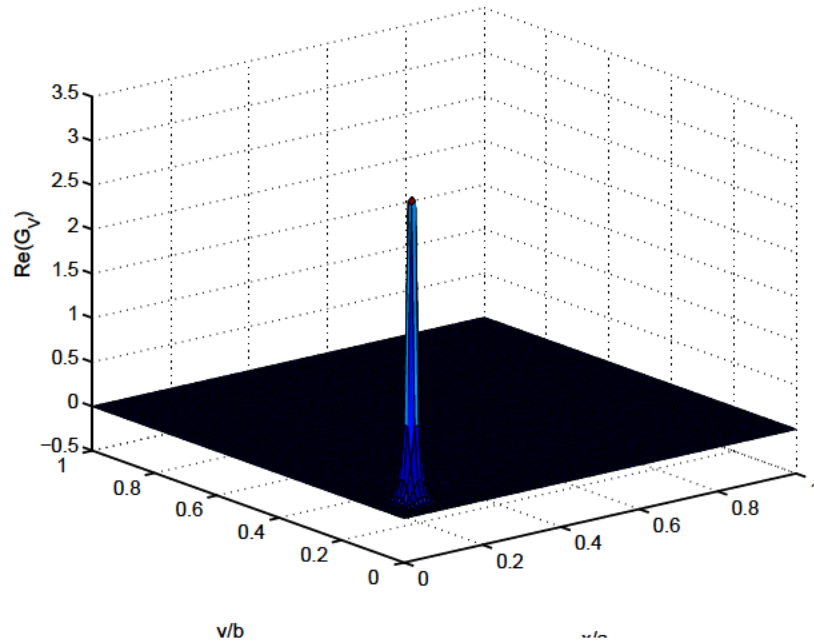


Figura 2.15: Parte real de la función de Green del potencial eléctrico escalar G_{cV} correspondiente a un dipolo excitación situado cerca de las paredes de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.

nuevo punto de entrada-salida. Si las funciones objetivo presentan fuertes variaciones, entonces los

valores de salida en los puntos de entrenamiento pueden ser muy diferentes a los valores en otros puntos no pertenecientes al conjunto de entrenamiento. En este caso, la red neuronal no calculará los valores de salida correctos ya que no posee una información completa sobre el comportamiento de las funciones objetivo. La presencia de la singularidad en las cercanías de la fuente es el mayor problema en la tarea de aproximación. En los capítulos siguientes se aplicarán diversas técnicas para extraer de forma eficaz la singularidad en la fuente. De este modo se disminuirán drásticamente las variaciones abruptas de las funciones de Green. Este hecho permitirá que la fase de entrenamiento asegure la consecución de redes neuronales que posean una apropiada capacidad de generalización.

2.5.2. Simetría de la cavidad

Debido a que vamos a tratar únicamente con cavidades en forma de hexaedro con caras rectangulares, la interfaz entre dos capas dieléctricas de la estructura multicapa puede dividirse en cuatro cuadrantes regulares. En la Figura 2.16 se aprecian los cuatro cuadrantes. Se ha situado un dipolo elemental o punto fuente en cada uno de los cuadrantes. Cada dipolo tiene asociado un punto observación que, aunque dibujado en el cuadrante propio del punto fuente, puede localizarse en cualquier punto del rectángulo:

$$\begin{aligned} 0 \leq x_i \leq a & \text{ Para } i=1,2,3,4 \\ 0 \leq y_i \leq b & \text{ Para } i=1,2,3,4 \end{aligned} \quad (2.77)$$

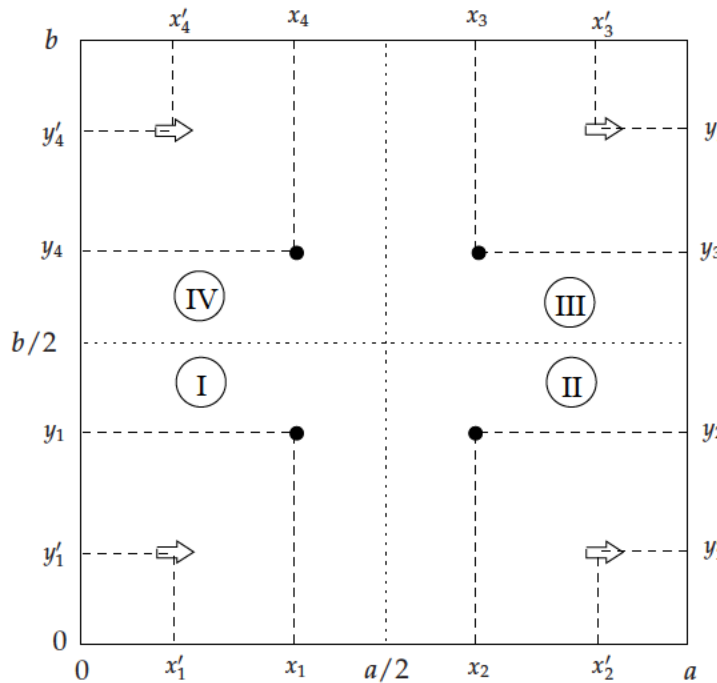


Figura 2.16: Interfaz entre capas dieléctricas en una estructura multicapa apantallada. Se muestran cuatro cuadrantes en los que se divide de forma regular la cavidad. En cada cuadrante aparece un dipolo excitación y un punto observación.

En la Figura 2.16 se observa la simetría de la estructura respecto de la recta vertical $x = a/2$ y de la recta horizontal $y = b/2$. Gracias a dicha simetría y a las características de las funciones sinusoidales que forman las series modales, podemos calcular las funciones de Green de cualquier punto fuente para cualquier punto observación a partir del correspondiente punto fuente simétrico del primer cuadrante. De este modo, para cada uno de los puntos fuente de los cuadrantes *I*, *II*, *III* y *IV* el punto simétrico del primer cuadrante (*I*) se evalúa como:

$$\text{Segundo cuadrante} \quad x'_1 = a - x'_2 \quad y'_1 = y'_2 \quad (2.78a)$$

$$\text{Tercer cuadrante} \quad x'_1 = a - x'_3 \quad y'_1 = b - y'_3 \quad (2.78b)$$

$$\text{Cuarto cuadrante} \quad x'_1 = x'_4 \quad y'_1 = b - y'_4 \quad (2.78c)$$

Asimismo dependiendo del cuadrante en el que se encuentre el punto fuente, el punto observación deberá sufrir una transformación para conseguir el punto observación asociado al punto fuente simétrico:

$$\text{Fuente en el segundo cuadrante:} \quad x_1 = a - x_2 \quad y_1 = y_2 \quad (2.79a)$$

$$\text{Fuente en el tercer cuadrante:} \quad x_1 = a - x_3 \quad y_1 = b - y_3 \quad (2.79b)$$

$$\text{Fuente en el cuarto cuadrante:} \quad x_1 = x_4 \quad y_1 = b - y_4 \quad (2.79c)$$

Para demostrar la simetría existente en las funciones de Green de la estructura multicapa se recurre a las funciones f y g definidas en la Tabla 2.1. Cada función de Green tiene asociado un tipo particular de producto de funciones f y g multiplicado por la correspondiente función de Green de espacio libre en el dominio espectral. A su vez cada función f y g está formada bien por la multiplicación de dos senos bien por la de dos cosenos dependientes de (k_{x_m}, x, x') o de (k_{y_n}, y, y') . Tomando el ejemplo particular de la función $f(k_{x_m}, x, x')$ que se utiliza en la definición de la función de Green G_{cV} :

$$f(k_{x_m}, x, x') = \sin(k_{x_m}x) \sin(k_{x_m}x') \quad (2.80)$$

Si el punto fuente se encuentra en el cuadrante segundo o tercero será obligatoria la aplicación de la transformación sobre la coordenada x para obtener la correspondiente abscisa tanto del punto fuente simétrico como del punto observación asociado. Aplicando dicha transformación y mediante simples transformaciones trigonométricas, la ecuación anterior se convierte en:

$$\begin{aligned} f(k_{x_m}, x, x') &= \sin(k_{x_m}(a - x_1)) \sin(k_{x_m}(a - x'_1)) \\ &= [\sin(k_{x_m}a) \sin(k_{x_m}x_1) - \cos(k_{x_m}a) \cos(k_{x_m}x_1)] \\ &\quad \cdot [\sin(k_{x_m}a) \sin(k_{x_m}x'_1) - \cos(k_{x_m}a) \cos(k_{x_m}x'_1)] \end{aligned} \quad (2.81)$$

Recuperando el valor del número de onda transversal:

$$\begin{aligned} \sin(k_{x_m} a) &= \sin\left(\frac{m\pi}{a} a\right) = 0 \quad \forall m \\ \cos(k_{x_m} a) &= \cos\left(\frac{m\pi}{a} a\right) = \begin{cases} 1 & \text{Para } m \text{ par} \\ -1 & \text{Para } m \text{ impar} \end{cases} = \text{signo}(m) \end{aligned} \quad (2.82)$$

Aplicando las ecuaciones de (2.82) en la ecuación (2.81) y teniendo en cuenta que aplicamos dos veces la función signo de m , se obtiene la función f de G_{cV} correspondiente a la pareja de puntos fuente y observación simétricos:

$$f(k_{x_m}, x, x') = \sin(k_{x_m} x_1) \sin(k_{x_m} x'_1) \quad (2.83)$$

De forma análoga se puede aplicar el mismo procedimiento para el producto de dos funciones seno que dependan de las ordenadas de los puntos fuente y observación. La transformación en las ordenadas es requerida si el punto fuente original se encuentra en el tercero y cuarto cuadrantes. En caso de pertenecer al primer o segundo cuadrantes no es necesario alterar los valores de las ordenadas. La transformación mostrada en las ecuaciones (2.79b, 2.79c) para las ordenadas permite obtener el punto simétrico adecuado para la función g , cuando está formada por el producto de dos funciones seno como en el caso de las funciones de Green G_{cV} y G_{cA}^{xx} .

Si la función f o g está compuesta por el producto de dos funciones coseno la transformación en las ordenadas o en las abscisas de las ecuaciones (2.79) es válida de igual forma. Tomamos la función f de la función de Green del potencial vectorial magnético G_{cA}^{xx} para mostrar el proceso:

$$f(k_{x_m}, x, x') = \cos(k_{x_m} x) \cos(k_{x_m} x') \quad (2.84)$$

Aplicando el cambio en las ordenadas y mediante simples transformaciones trigonométricas:

$$\begin{aligned} f(k_{x_m}, x, x') &= \cos(k_{x_m} (a - x_1)) \cos(k_{x_m} (a - x'_1)) \\ &= [\cos(k_{x_m} a) \cos(k_{x_m} x_1) + \sin(k_{x_m} a) \sin(k_{x_m} x_1)] \\ &\quad \cdot [\cos(k_{x_m} a) \cos(k_{x_m} x'_1) + \sin(k_{x_m} a) \sin(k_{x_m} x'_1)] \end{aligned} \quad (2.85)$$

Empleando de nuevo las ecuaciones de (2.82) demostramos que la función f original se puede calcular utilizando los puntos simétricos ubicados en el primer cuadrante:

$$f(k_{x_m}, x, x') = \cos(k_{x_m} x_1) \cos(k_{x_m} x'_1) \quad (2.86)$$

Del mismo modo se puede demostrar que la transformación es apta cuando el producto de funciones tipo coseno depende de las ordenadas de los puntos fuente y observación. Todas las funciones

de Green están formadas por funciones f y g y por una función espectral cuyo valor no depende ni de x ni de y . Por tanto se pueden calcular las funciones de Green de una fuente en cualquier punto de la cavidad recurriendo al correspondiente punto simétrico. Esta característica va a permitir limitar el espacio de entrenamiento correspondiente a los puntos fuente, al primer cuadrante de la interfaz dieléctrica. De este modo los valores de entrada de la red neuronal correspondientes a las coordenadas de los puntos fuente pertenecerán siempre al primer cuadrante. Cuando sea necesario obtener los valores de salida de la red neuronal de un punto fuente situado fuera del primer cuadrante se recurrirá a las transformaciones de (2.78,2.79) para obtener el valor de entrada apropiado.

2.6. Conclusiones

En este Capítulo se han explicado las técnicas empleadas para calcular las funciones de Green de los potenciales relevantes del medio multicapa encapsulado. Estas funciones en el dominio espacial están formadas por series modales que implican funciones sinusoidales oscilantes. Por ello, la suma directa de las series converge lentamente. Existen diferentes métodos para acelerar las sumas modales. En este trabajo se utilizan los procedimientos descritos en [Álvarez Melcón, 1998] que permiten una eficiente evaluación de las series modales. El método general consta de dos técnicas que se emplean en diferentes zonas del espacio en el que se sitúan los puntos observación. Así, cerca de la fuente se aplica la fórmula de Poisson para la parte estática de las funciones de Green. Esta fórmula posibilita la sustitución del problema original por una distribución de imágenes espaciales cuyos términos convergen rápidamente. La parte dinámica formulada en el dominio espectral no presenta problemas de convergencia, de modo que se puede sumar directamente. Lejos de la fuente se aplica la técnica de suma por partes. Este procedimiento permite la expresión del sumatorio doble original en función de sumatorios de funciones sinusoidales que resultan ser analíticos. Para cada estructura multicapa apantallada se define un umbral que determina los límites de utilización de cada uno de los métodos. Este umbral se fija con el objetivo de minimizar el tiempo de evaluación de las funciones de Green en la estructura de trabajo.

Las funciones de Green de los potenciales pueden ser empleadas en la resolución de la Ecuación integral (IE) mediante el Método de los momentos (MoM). La formulación de la IE en el dominio espacial presenta ciertas ventajas frente a las formulaciones espectrales. Sin embargo, a pesar de los avances realizados en la aceleración de las funciones de Green en el dominio espacial, éstas ralentizan la resolución de la IE. Por tanto, si se desea emplear una técnica precisa de análisis electromagnético en nuevas técnicas de diseño, es obligatorio que las funciones de Green sean calculadas rápida y eficazmente.

En la presente tesis, las funciones de Green son aproximadas por redes neuronales. De este modo, los valores de salida de las redes neuronales sustituirán a los valores obtenidos con los métodos numéricos descritos. En este Capítulo se han mostrado dos propiedades de las funciones de Green que van a incidir decisivamente en la tarea de aproximación. Por un lado, la singularidad en la fuente provoca un crecimiento muy abrupto de las funciones de Green en la zona espacial situada en torno a la fuente. Este hecho imposibilita de forma práctica la aproximación por medio de redes neuronales. Por ello, en el siguiente Capítulo se desarrollan diferentes mecanismos para eliminar de manera eficaz la singularidad de la fuente. En segundo lugar, las funciones de Green en una estructura en

forma rectangular son simétricas respecto de los ejes centrales que dividen en partes iguales al plano transversal. Gracias a esta característica, se puede limitar el conjunto de entrenamiento de puntos fuente a un cuadrante de dicho plano. Así, los conjuntos de entrenamiento pueden ser reducidos, disminuyendo el tiempo de diseño de las redes neuronales. Este hecho redundará en un aumento de la eficiencia de la aproximación de las funciones de Green con redes neuronales.

Capítulo 3

Método neuronal de análisis de circuitos apantallados multicapa basado en Redes neuronales de funciones de base radial

3.1. Introducción

En el Capítulo 2 se comprobó que la evaluación de las funciones de Green en el medio multicapa apantallado es una tarea compleja. Las series infinitas que componen las funciones de Green de los potenciales deben ser adecuadamente tratadas para realizar su suma correctamente. Para acelerar la evaluación de las mencionadas series es posible recurrir a diversas transformaciones tanto en el dominio espacial como en el dominio espectral. Sin embargo, pese a los avances realizados, la estimación de las funciones de Green es la labor que consume más tiempo en la resolución de la Ecuación integral (IE) mediante el Método de los momentos (MoM). Por tanto, si los valores de las funciones de Green fueran obtenidos más velozmente, entonces el análisis circuital sería acelerado en gran medida. En el presente trabajo, la aceleración no se lleva a cabo mediante técnicas que operen sobre las series modales de las funciones de Green. En lugar de actuar directamente sobre las funciones de Green, se emplean redes neuronales que sustituyen a las propias funciones de Green durante la evaluación de la IE.

Las redes neuronales son capaces de aprender las relaciones no lineales que existen en un conjunto de datos de entrada-salida multidimensionales. El conocimiento adquirido se puede utilizar principalmente en tres aplicaciones diferentes. La clasificación de los datos de salida en distintos patrones o clases, la predicción de datos de salida futuros en series temporales y finalmente la descripción de las variaciones de los datos de salida respecto de los datos de entrada. Este último caso se denomina tarea de regresión y equivale a la aproximación mediante las redes neuronales de la función matemática o proceso que representan los datos. En nuestro problema, las redes neuronales seleccionadas deberán aproximar las funciones de Green de los potenciales relevantes del medio. La utilidad de las redes neuronales estriba en la rapidez con la que se calculan sus valores de salida. Por lo tanto, una vez finalizada la fase de aprendizaje o entrenamiento, las redes neuronales acelerarán

la resolución de la IE gracias a que las lentas técnicas de evaluación de las series infinitas de las funciones de Green serán sustituidas por el rápido procedimiento de cálculo de las salidas neuronales.

La habilidad de las redes neuronales para capturar las relaciones y estructura de los datos de una forma no lineal junto con su rapidez de respuesta ha sido explotada en diversos problemas electromagnéticos [Christodoulou y Georgipoulos, 2001]. Los primeros trabajos con redes neuronales consistieron en la creación de modelos de ciertos componentes y circuitos de microondas para los que no existían modelos teóricos adecuados que los caracterizaran [Wang y Zhang, 1997, Vai y Prasad, 1995, Horng et al., 1993]. Las redes neuronales también fueron aplicadas a la optimización de circuitos de microondas; podemos citar como trabajos pioneros a [Watson y Gupta, 1997, Zaabab et al., 1995]. Después de que las redes neuronales probaran su habilidad en la caracterización, modelado, optimización y diseño de componentes y circuitos de microondas, llegó el momento del desarrollo de programas de diseño asistido por computador ("Computer aided design" o CAD según las siglas inglesas) como los desarrollados en [Wang et al., 1998] y [Creech et al., 1997] que incluían los avances realizados hasta ese momento.

Asimismo, se han realizado diversas aportaciones en el campo del modelado de circuitos impresos en un medio multicapa. Los trabajos en este campo emplean las redes neuronales en la técnica de la IE resuelta con el MoM. En [Soliman et al., 2003] se utilizaron redes neuronales para calcular los coeficientes del método de las imágenes complejas ("Discrete complex image method" o DCIM según las siglas inglesas) [Yang et al., 1992]. Concretamente, se emplearon Redes neuronales de funciones de base radial (RBFNN). Este método neuronal se empleó para analizar, en un ancho de banda determinado, un dipolo eléctrico localizado en la parte superior de un sustrato dieléctrico. Las redes neuronales también han sido empleadas en técnicas elaboradas en el dominio espectral. Así, en [Mishra y Patnaik, 1999] el tradicional Perceptrón multicapa aproxima directamente las integrales, fruto de aplicar el MoM, que relacionan los campos eléctricos superficiales y las distribuciones de corrientes en una estructura microtira de dimensiones transversales infinitas. Debido a que se utiliza una formulación en el dominio espectral la técnica se denominó método neuro-espectral. Este método se aplicó en distintos problemas, como el cálculo de la frecuencia de resonancia de un resonador microtira [Mishra y Patnaik, 1999], la estimación de la impedancia de una antena microtira y el diseño de una antena rectangular de tipo microtira [Mishra y Patnaik, 2003]. La aceleración de la IE, gracias a la aproximación de los coeficientes de la matriz de momentos por medio de redes neuronales, también ha sido utilizada en el dominio espacial. En [Soliman et al., 2004] aquellos términos de la matriz de acoplos, surgida de la aplicación del MoM, correspondientes a puntos observación situados lejos de la fuente fueron aproximados por Redes neuronales de funciones de base radial (RBFNN). La limitación del método neuronal era debida a la incapacidad de las redes neuronales para aproximar las fuertes variaciones que se producen en las cercanías de la fuente. En consecuencia, la mencionada técnica debía usar la evaluación directa de los términos correspondientes a aquellos puntos emplazados cerca de la fuente con la consiguiente pérdida de eficiencia.

El enfoque del presente trabajo es novedoso por varias razones. En lugar de emplear una red neuronal para aproximar los términos de la matriz de momentos, las redes neuronales se encargan de aproximar directamente las funciones de Green. Así, los valores de salida obtenidos con las redes neuronales se utilizarán en el llenado de la matriz de momentos. De este modo, el método neuronal es mucho más versátil, ya que cualquier circuito impreso podrá ser analizado con precisión en la es-

estructura de trabajo. Además, las redes neuronales han sido empleadas por primera vez en un entorno multicapa totalmente apantallado. El análisis de circuitos en este medio es especialmente complicado debido a las restricciones impuestas por las dimensiones finitas de la cavidad metálica. Las redes neuronales pueden así demostrar de forma clara su validez para acelerar el análisis de los circuitos de microondas encapsulados. Para lograr este objetivo por medio de la aproximación de las funciones de Green se deben superar diversos problemas. El principal obstáculo radica en las rápidas variaciones que ocurren en determinadas circunstancias. Cerca de la singularidad en la fuente, para cualquier estructura y frecuencia, las funciones de Green exhiben rápidas variaciones que deben ser completamente representadas para su correcta aproximación. En esta tesis se han aplicado dos técnicas para suavizar la evolución de las funciones de Green cerca de la fuente en particular y en todo el espacio en general. La primera técnica está basada en la multiplicación de las funciones de Green por la distancia entre fuente y punto observación (ρ). Este procedimiento anula el término singular en la fuente. La segunda técnica actúa sobre el número de capas de influencia k (ver Figura 2.5) sumadas en la serie de imágenes espaciales. Este método permite la completa eliminación de la singularidad. De esta forma se obtiene una función de Green de muy lenta variación. Además, la singularidad es recuperada sin adición ni incremento de error alguno por lo que la reconstrucción de las funciones de Green a partir de los valores neuronales es muy precisa. Un segundo escenario en el que aparecen fuertes variaciones en las funciones de Green surge en presencia de una frecuencia de resonancia propia de la estructura apantallada. En este caso es necesario realizar una adecuada división del espacio de entrada de la red neuronal en la coordenada frecuencial para garantizar un entrenamiento correcto en las cercanías de dicha frecuencia de resonancia.

A pesar de la eliminación de la singularidad en torno al punto fuente, la aproximación de las funciones de Green es complicada a causa de la gran cantidad de puntos de entrenamiento necesarios para representar completamente sus variaciones en toda la cavidad. Por ejemplo, las redes neuronales deben reproducir el comportamiento de las funciones de Green en las proximidades de las paredes metálicas para expresar correctamente las condiciones de contorno del problema. Asimismo, se deben representar las variaciones de las funciones de Green a lo largo del ancho de banda de trabajo para permitir el estudio de circuitos impresos en diversas frecuencias. El gran tamaño del conjunto de entrenamiento representa un inconveniente fundamental ya que limita el número mínimo de neuronas necesarias para alcanzar un determinado nivel de error. En nuestro problema es de máximo interés minimizar el número de neuronas en las redes neuronales para minimizar a su vez el tiempo de análisis circuital. Si empleamos un número de neuronas demasiado reducido en un conjunto muy amplio de datos es altamente probable que el error en la aproximación sea tal que conduzca a un análisis incorrecto de los circuitos de microondas. Para evitar este obstáculo se ha desarrollado una técnica de división del espacio de entrada que permite una representación apropiada de las funciones de Green. La división del plano transversal en el que se sitúan los puntos fuente y observación en diversas zonas posibilita la correcta aproximación tanto cerca como lejos de la fuente. También permite el cumplimiento de las condiciones de contorno en las paredes metálicas. La división en zonas frecuenciales permite la correcta aproximación de las funciones de Green en las cercanías de las frecuencias de resonancia. Mediante la completa división del espacio de entrada se obtiene una serie de redes neuronales que logran una aproximación eficiente, ya que alcanzan un nivel de error pequeño con un número reducido de neuronas.

Cada tipo de red neuronal presenta una serie de parámetros propios que deben ser calculados durante la fase de entrenamiento. La Red neuronal de funciones de base radial (RBFNN) ha sido seleccionada para aproximar las funciones de Green. Las funciones de base radial utilizadas son gaussianas caracterizadas por una varianza y un centro o centroide. La justificación del uso de esta red neuronal se basa en su facilidad de entrenamiento y fiabilidad en tareas de regresión [Haykin, 1999]. El método de entrenamiento seleccionado ha sido el de Mínimos cuadrados ortogonales ("Orthogonal Least Squares" u OLS según las siglas inglesas). Al principio del Capítulo se detalla el estudio realizado para seleccionar este método de entrenamiento en lugar de otras técnicas alternativas. En el método OLS es necesario fijar la varianza de las neuronas gaussianas con anterioridad al comienzo del entrenamiento. Una adecuada estimación de la varianza es fundamental para conseguir un funcionamiento eficiente. Por ello, se ha diseñado un algoritmo de búsqueda de la mejor varianza que incrementa la capacidad de aproximación de las RBFNNs. El algoritmo de búsqueda de varianza actúa en combinación con el mecanismo de división en zonas del espacio de entrada. Los mecanismos anteriores permiten la generación de conjuntos de entrenamiento reducidos que requerirán tiempos de entrenamiento poco prolongados. Aunque el entrenamiento es una fase inicial que solo debe ser realizada una vez, es aconsejable acortar el tiempo destinado a esta tarea. Por este motivo, se ha operado sobre la implementación del algoritmo OLS para conseguir un tiempo de entrenamiento moderado.

Finalmente, para expresar de manera más fiel el comportamiento de las funciones de Green cerca de la fuente se han generado zonas de puntos observación de tipo circular. Cerca de la fuente, sobre todo cuando ésta se localiza lejos de las paredes metálicas, las funciones de Green poseen simetría esférica. Los resultados muestran una gran mejora del entrenamiento de las RBFNNs cuando todos los mecanismos actúan conjuntamente. De este modo, se han podido analizar circuitos encapsulados multicapa de cierta complejidad de forma rápida y fiable.

La estructura del presente Capítulo se compone de nueve secciones. En la siguiente Sección se explican las razones que han conducido a la elección de las RBFNNs como red neuronal en el problema propuesto. En la Sección 3.3 se exponen tres algoritmos de entrenamiento ampliamente utilizados en el diseño de las RBFNNs. El Capítulo continúa en la Sección 3.4 con la explicación del primer mecanismo de extracción de la singularidad basado en la multiplicación de las funciones de Green por la distancia (ρ). Las funciones de Green desprovistas de la singularidad en la fuente pueden ser aproximadas por una red neuronal. En la Sección 3.5 se comprueba el rendimiento de los diferentes métodos de entrenamiento mostrados en la Sección 3.3. Una vez seleccionado el método de entrenamiento más eficiente se procede en la Sección 3.6 a la comparación del rendimiento de la RBFNN con el de otras redes neuronales como el Perceptrón multicapa y las Máquinas de vectores soporte. En la Sección 3.7 se expone el primer algoritmo de división del espacio de entrada de la RBFNN. En la Sección 3.8 el algoritmo anterior es mejorado y completado con el mecanismo que busca de forma iterativa la mejor varianza gaussiana para cada zona espacial. Finalmente, en la Sección 3.9 se explican dos procedimientos que contribuyen a una importante mejora en la precisión y eficiencia del método neuronal. En primer lugar, se explica la técnica de cancelación de la singularidad basada en la extracción de capas de influencia. En segundo lugar, se aplican zonas circulares en lugar de cuadradas en la región cercana a la fuente. Además, se utiliza una versión mejorada del algoritmo de entrenamiento OLS que permite una reducción en el tiempo de entrenamiento. En cada una de

las técnicas empleadas se analizan diferentes filtros encapsulados con la inclusión de medidas para validar las simulaciones realizadas. Finalmente, en la Sección 3.10, se exponen las conclusiones.

3.2. Selección de la red neuronal

La red neuronal debe cumplir una serie de requisitos para aproximar las funciones de Green de los potenciales con el grado de precisión deseado. En primer lugar, debe ser apta para realizar tareas de regresión, es decir, para aproximar funciones o procesos. En segundo lugar, el algoritmo de entrenamiento de la red neuronal debe posibilitar que el número de neuronas empleado en cada conjunto de datos sea pequeño, a fin de conseguir que la matriz de momentos sea llenada rápidamente. En tercer lugar, la red neuronal y su algoritmo de entrenamiento deben ser capaces de trabajar con conjuntos de datos relativamente grandes. La división en zonas limita la cantidad de datos en cada conjunto de entrenamiento. A pesar de ello, debido al interés de aproximar las funciones de Green en toda la cavidad y a lo largo del ancho de banda de trabajo es necesario recurrir a conjuntos de datos relativamente grandes. En resumen, se seleccionará aquella red neuronal y método de entrenamiento competentes para producir redes neuronales compactas sobre conjuntos de datos de tamaño relativamente grande. Además, es idóneo que la red neuronal emplee un tiempo de entrenamiento reducido, de forma que no convierta en poco práctico el método neuronal de análisis circuital.

A continuación se describe el funcionamiento de diversos tipos de redes neuronales de extendido uso en tareas similares al problema tratado. Asimismo, se exponen las razones que han llevado a la decisión de descartar ciertas redes neuronales y seleccionar finalmente la Red neuronal de funciones de base radial (RBFNN).

3.2.1. Perceptrón multicapa (MLP)

El Perceptrón multicapa (MLP) es la clase de red neuronal más utilizada en problemas electromagnéticos [Zhang et al., 2003]. Esta red neuronal está compuesta por una capa de neuronas de entrada, una o varias capas ocultas o intermedias de neuronas y una capa de neuronas de salida. El número de neuronas de la capa de entrada coincide con la dimensión del espacio de entrada, así como el número de neuronas de salida coincide con la dimensión del espacio de salida. En cambio, el tamaño de cada una de las capas ocultas no depende de ninguna dimensión. Cada capa oculta puede poseer un número diferente de neuronas como se aprecia en la Figura 3.1. La entrada de cada una de las neuronas intermedias se calcula mediante la suma de las salidas de las neuronas de la capa anterior a las que está conectada, multiplicadas por los correspondientes pesos. Este hecho implica que la entrada a una neurona es el producto escalar entre un vector de pesos y un vector de salidas de otras neuronas. Así, cada salida neuronal es ponderada por un determinado peso que indica su importancia en la red neuronal.

Cada neurona está caracterizada por una función de activación (φ) no lineal que determina el valor de salida neuronal. Dos de las funciones más empleadas son la función logística o sigmoideal y la función tangente hiperbólica. Las funciones anteriores poseen un valor de saturación a partir del cual el valor de salida permanece constante. Su forma provoca que el MLP realice una aproximación global sobre toda la función objetivo en una tarea de regresión.

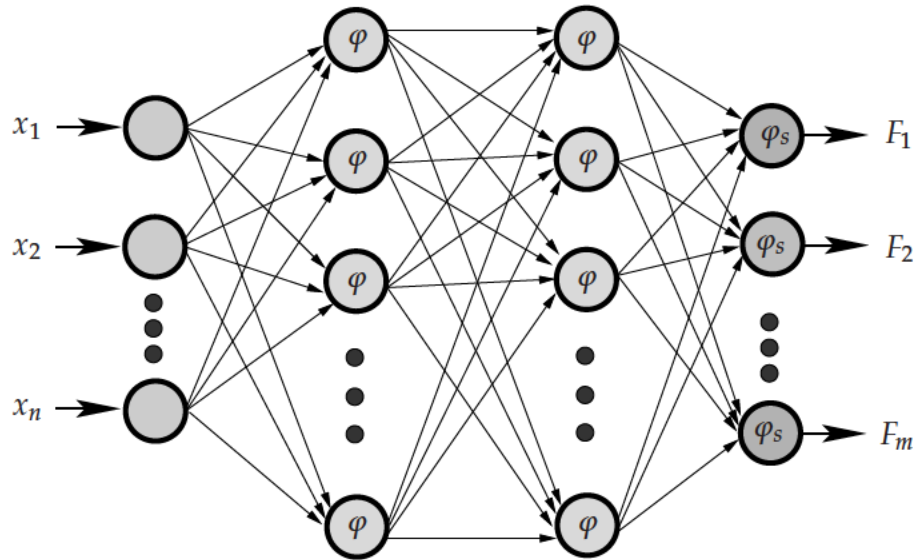


Figura 3.1: Estructura de un Perceptrón multicapa. Las funciones φ de las capas ocultas pueden ser de igual o diferente tipo que las funciones de la capa de salida φ_s . El espacio de entrada posee dimensión n y el espacio de salida dimensión m .

El método de entrenamiento más aplicado y extendido en este tipo de redes neuronales es el mecanismo de retro-propagación del error [Haykin, 1999]. Es un método que opera en dos fases. En la primera fase una nueva entrada se transmite por la red hasta la salida. En esta última capa, la salida de la red se compara con el valor deseado y se calcula un determinado tipo de error. Dicho error va a producir una regla de corrección que se propaga desde la salida hacia la entrada en la segunda fase del entrenamiento. Durante la propagación del error los pesos que conectan las diferentes neuronas son alterados para minimizar el error en la salida. La red neuronal MLP satisface el teorema de aproximación universal [Hornik et al., 1989] y [Cybenko, 1989]. Esta propiedad implica que un MLP puede aproximar cualquier función acotada con cualquier grado de exactitud. Este teorema señala meramente la existencia de un MLP determinado que aproxima una función dada, pero no informa sobre la estructura exacta de dicha red neuronal. De hecho, un MLP de una sola capa es suficiente para constituir un aproximador universal. Sin embargo, el MLP de una sola capa no tiene por qué constituir el MLP más eficiente en términos de tiempo de entrenamiento o capacidad de generalización. Además, a pesar de satisfacer de forma teórica el mencionado teorema, el MLP se construye en la práctica a partir de un grupo limitado de datos. Así, los datos pueden no ser suficientes para que el MLP aproxime correctamente toda la función en el espacio acotado. A esta situación se le denomina sobre-entrenamiento y se detecta debido al aumento del error de validación o test respecto del error de entrenamiento. En el sobre-entrenamiento las neuronas de la red neuronal “memorizan” los datos de entrenamiento. Por consiguiente, no son capaces de generalizar a partir de los datos ya conocidos. Este problema se suele solucionar aumentando el número de puntos de entrenamiento. Si el error en el conjunto de validación es similar al de entrenamiento pero éste no es pequeño, existe una situación de bajo-entrenamiento. La solución consiste en la adición de nuevas neuronas. En el campo del diseño y modelado de circuitos de microondas con MLP se han implantado diversos métodos para evitar y solucionar estos errores [Devabhaktuni et al., 2001]. Estas dos situaciones indeseables aparecen en el diseño de otras redes neuronales como las RBFNNs.

Los MLP presentan características positivas respecto a su capacidad de aproximación de funciones. A pesar de que el MLP de varias capas puede ser más eficiente que un MLP formado por dos capas, éste último es el comúnmente empleado en aplicaciones prácticas. No obstante, los MLP presentan desventajas frente a las RBFNNs en la tarea de aproximación de las funciones de Green. Su estructura es más compleja, incluso en el caso de que estén compuestos por una única capa intermedia de neuronas. Además, el tiempo de entrenamiento es mucho mayor en un MLP que en una RBFNN [Lippman, 1989] y [Verma, 1995]. Como ya se comentó en la introducción, si el entrenamiento es muy prolongado, a pesar de constituir un periodo inicial, puede llegar a transformar en poco práctico el método de análisis neuronal. Los MLP no superan en general el rendimiento de las RBFNNs en tareas de regresión ni clasificación. Las RBFNNs, especialmente si son entrenadas con un método supervisado como el OLS, alcanzan niveles de error menores que los conseguidos con el MLP.

Esta minusvalía en las prestaciones del MLP frente a la RBFNN es debida a distintos factores. Por un lado los pesos de un MLP son generalmente inicializados de forma aleatoria. Es difícil encontrar un conjunto de pesos iniciales apropiados para un problema determinado. El rendimiento del MLP dependerá de la elección primaria de pesos, de forma que una mala inicialización provocará una lenta reducción del error de aproximación. Por otro lado, el método de retro-propagación altera los pesos mediante algoritmos basados en la evaluación del gradiente del error. Este hecho, en conjunción con la inicialización aleatoria, puede provocar que el algoritmo de optimización quede atrapado en un mínimo local. En este caso, por más iteraciones del algoritmo de entrenamiento que se realicen no se producirá una mejora de la capacidad de aprendizaje del MLP. Finalmente, los MLP presentan una menor tolerancia a fallos en comparación con las RBFNNs. Este inconveniente es especialmente importante si se realiza una implementación hardware. Si una neurona sufre daño, debido al carácter global de la aproximación, degradará el nivel de error a lo largo de todo el espacio de trabajo. El daño se propagará por todas las neuronas. En cambio, una RBFNN realiza una aproximación de carácter local. Una neurona solo afectará a su zona de influencia limitando la degradación del error.

3.2.2. Máquinas de vectores soporte (SVM)

Las Máquinas de vectores soporte (SVM) son un tipo de redes neuronales de una única capa desarrolladas durante los años noventa [Vapnik, 1992], [Boser et al., 1992], [Cortes y Vapnik, 1995] y [Vapnik et al., 1997]. Las SVM fueron en un principio diseñadas para construir un hiperplano que separase los diferentes patrones o clases en problemas de clasificación. En el caso de patrones separables linealmente una SVM construirá el hiperplano que logra una separación óptima. Cuando las clases no son separables de forma lineal, el hiperplano que genera la SVM minimiza la probabilidad de error de clasificación gracias a la aplicación del método de minimización del riesgo estructural. Este mecanismo se fundamenta en la minimización de la dimensión de Vapnik-Chervonenkis (VC). El error de generalización de una red neuronal depende del error de entrenamiento y de la dimensión VC. Mediante la minimización de dicha dimensión las SVM consiguen reducir la cantidad de patrones mal clasificados tanto en el conjunto de entrenamiento como en el conjunto de test.

La arquitectura de las SVM es resultado de la aplicación del teorema de Cover de separación de patrones [Cover, 1965]. De acuerdo con este teorema, un conjunto de patrones o clases, originalmente

no separables linealmente, son separables de forma lineal con gran probabilidad si el espacio de entrada es proyectado o transformado en otro espacio multidimensional. La probabilidad de éxito en la separación de patrones crece conforme aumenta la dimensión del nuevo espacio multidimensional respecto a la dimensión del espacio de entrada. En una SVM cada coordenada del nuevo espacio multidimensional se calcula por medio de una neurona. Los datos transformados en el espacio multidimensional son separados linealmente, de forma que cada coordenada o salida neuronal es multiplicada por un peso. Por tanto, la estructura de una SVM estará compuesta por una única capa de neuronas que generan una transformación no-lineal y una capa de salida en la que se realiza la transformación lineal (ver Figura 3.2). El espacio multidimensional en el que se produce la separación lineal se denomina también espacio oculto o espacio de la capa oculta de neuronas.

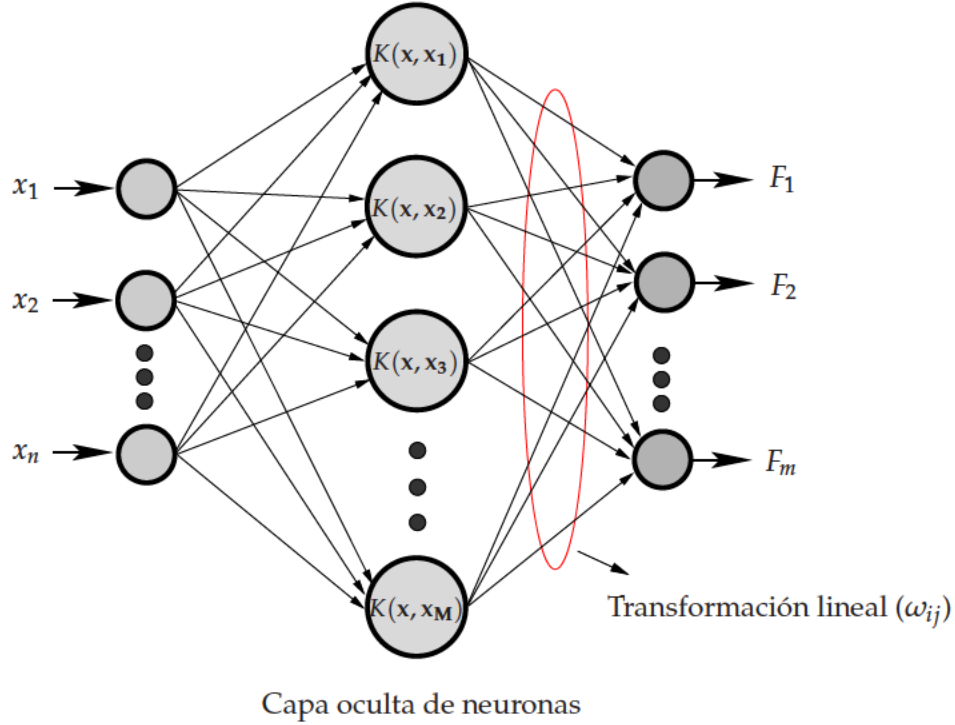


Figura 3.2: Estructura de una Máquina de vectores soporte (SVM). Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. La capa oculta realiza una transformación no lineal. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida.

Al igual que en el Perceptrón multicapa cada neurona de una SVM está caracterizada por una función matemática. En una SVM la función no-lineal de la neurona debe ser el producto interior de una determinada función. El teorema de Mercer indica las condiciones que debe cumplir la función no-lineal para que sea considerada un producto interior adecuado o función de kernel [Mercer, 1909]. El producto interior utiliza un dato de entrada y un vector proveniente del conjunto de datos de entrenamiento que denominamos vector soporte. La operación que realiza una SVM se expresa matemáticamente como:

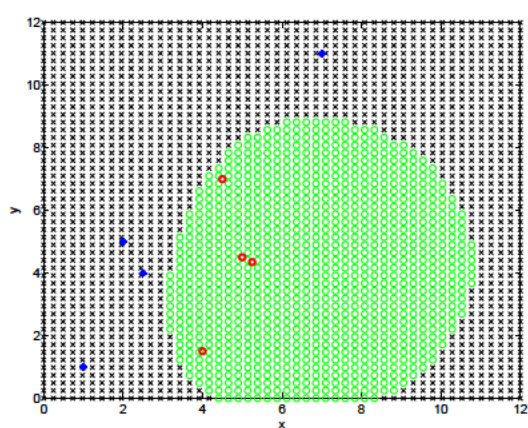
$$F_j(\mathbf{x}) = \sum_{i=1}^M \omega_{ij} K(\mathbf{x}, \mathbf{x}_i) \quad (3.1)$$

donde M es el número de neuronas, ω_{ij} es el peso de la neurona i -ésima para la salida j -ésima, \mathbf{x}_i el vector soporte i -ésimo y $K(\cdot)$ es la función propia de la SVM. El diseño de una SVM implica el cálculo de los vectores soporte y los pesos pertenecientes a la capa de salida. Dicho cómputo se lleva a cabo mediante la maximización de una función objetivo sujeta a ciertas restricciones. El problema de optimización cuadrática se resuelve aplicando el método de los multiplicadores de Lagrange. Aquellos multiplicadores de Lagrange distintos de cero indican la presencia de un vector soporte. Finalmente, los pesos se calculan con los multiplicadores de Lagrange, los vectores soporte y los datos de salida apropiados.

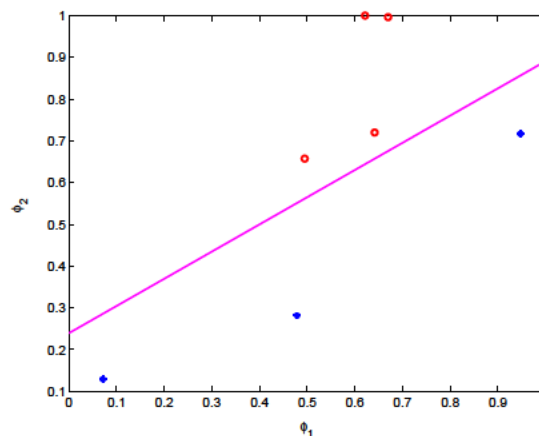
En el campo de las microondas encontramos pocos trabajos que hagan uso de las redes neuronales de tipo SVM. Sin embargo, una SVM puede ser empleada en problemas de regresión mediante la definición adecuada de la función objetivo que debe ser maximizada. El problema de optimización en el caso de la aproximación de una función o proceso es más complicado que el correspondiente problema en una tarea de clasificación. Existen diversas técnicas que permiten una reducción sustancial de la carga computacional asociada al problema de optimización cuadrática. Podemos citar el algoritmo de optimización secuencial mínimo ("Sequential Minimal Optimization" o SMO según las siglas inglesas) [Flake y Lawrence, 2002]. Entre los métodos más eficientes se encuentra el algoritmo SVM-Light descrito en [Joachims, 1999] y [Joachims, 2002], que alcanza una rapidez dos veces superior al algoritmo SMO. La técnica SVM-Light ha sido aplicada con éxito a problemas de regresión tanto en su implementación original como en posteriores versiones [Collobert y Bengio, 2001]. A pesar de la mejora en la eficiencia del cálculo de los parámetros de la SVM, todas las técnicas anteriores generan una elevada cantidad de vectores soporte, es decir, de neuronas. En el problema de la aproximación de las funciones de Green no sólo es importante lograr un grado de precisión elevado sino también mantener un número limitado de neuronas. Solo así se conseguirá reducir el tiempo de análisis de circuitos apantallados. Un número elevado de neuronas ocasionará que el método de análisis neuronal resulte poco atractivo debido a la poca ganancia en el tiempo de cálculo.

3.2.3. Redes neuronales de funciones de base radial (RBFNN)

Las Redes neuronales de funciones de base radial (RBFNN) son un tipo de red neuronal apta tanto para tareas de clasificación como de regresión. Al igual que las SVM su estructura está fundamentada en el teorema de Cover de separación de patrones [Cover, 1965]. Este teorema asegura que un problema de clasificación complejo será linealmente separable con mayor probabilidad si se realiza una proyección no lineal en un espacio de dimensión mayor que el espacio de partida. La proyección no-lineal se produce en la capa oculta de neuronas. En las RBFNNs se realiza una transformación lineal en la capa de salida de la misma forma que en las SVM. De hecho, en los casos en los que la función de base radial utilizada en la RBFNN cumple el teorema de Mercer se puede emplear la teoría de construcción de las SVM para diseñar la red neuronal. Este es el caso de las funciones de base radial de tipo gaussiano. Desde este punto de vista, las RBFNNs de tipo gaussiano son un caso particular de SVM, ya que éstas pueden emplear otros tipos de funciones como polinomios o funciones de tipo sigmoidal. Asimismo, debido a que existen distintas funciones de base radial que pueden servir para construir una RBFNN, el método de construcción aplicado en las SVM puede verse como un tipo de entrenamiento particular de RBFNN gaussianas.

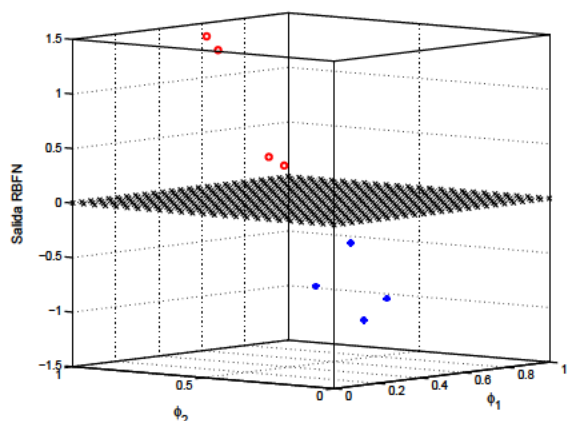


(a) Ejemplo de dos clases (asteriscos y círculos) no separables linealmente. La RBFNN genera dos clases de puntos (cruces y cuadrados).

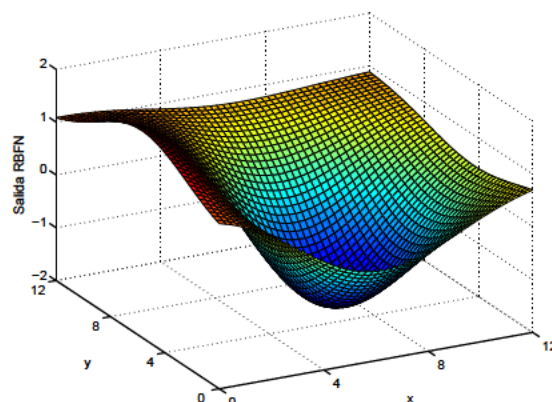


(b) Patrones en el espacio de la capa oculta de la RBFNN. Los patrones son separables linealmente.

Figura 3.3: Patrones en el espacio original y en el espacio de la capa oculta de la RBFNN.



(a) Salida de la RBFNN y plano de separación de patrones en el espacio de la capa oculta.



(b) Superficie generada por la RBFNN en el espacio original.

Figura 3.4: Salida de la RBFNN en el espacio oculto (a) y en el espacio original (b).

En el ejemplo de la Figura 3.3(a) se observa cómo una RBFNN es capaz de separar patrones de forma no lineal. Las dos clases de puntos no son separables linealmente en el espacio original ($x - y$). Una RBFNN compuesta por dos neuronas es capaz de generar una curva de separación no lineal que diferencia ambas clases. El espacio de la capa oculta de la RBFNN es bidimensional al igual que el espacio original. Sin embargo, en este nuevo espacio ($\phi(x) - \phi(y)$) los patrones proyectados sí son separables linealmente. La línea de separación de la Figura 3.3(b) se genera gracias a los pesos de la capa de salida. En este caso no ha sido necesario recurrir a un espacio de capa oculta de dimensión superior al espacio original debido a la sencillez de la distribución de patrones. Finalmente, se aprecia en la Figura 3.4(a) los valores de la salida de la RBFNN para cada uno de los patrones. El valor positivo o negativo de la red neuronal indica la presencia de uno u otro patrón. Asimismo, se muestra el plano de valores iguales a cero que constituye el plano de separación entre clases. Si el espacio de la capa oculta es de dimensión superior a dos entonces las clases son separadas por un hiperplano.

Las RBFNNs no sólo sirven para diferenciar patrones que no son separables en un determinado espacio. El procesamiento de los datos de entrada en una RBFNN se asimila a la construcción de una superficie multidimensional que se puede emplear en problemas de interpolación. Por ejemplo, en la Figura 3.4(b) se observa la superficie generada por la red neuronal que separa los patrones de la Figura 3.3(a). La forma de la superficie toma un valor mayor que cero en los puntos pertenecientes a un patrón y un valor menor que cero en los puntos del otro patrón. Si deseamos utilizar una RBFNN en un problema de aproximación de una función, entonces deberá ser entrenada con valores de salida continuos, no discretos. De este modo, el problema de regresión en una RBFNN consiste en la elaboración de una superficie que se adapte de forma precisa a la curva multidimensional que representan los datos de salida del conjunto de entrenamiento. Cuando se presenta un nuevo dato en la entrada, la RBFNN realiza una interpolación sobre la superficie construida durante el proceso de entrenamiento como mejor aproximación de la curva original. La capacidad de generalización es por tanto la habilidad de interpolación de la RBFNN. Este enfoque fue aplicado por primera vez a una red neuronal en [Broomhead y Lowe, 1988]. Una función de base radial es aquella en la que el argumento de la función depende de la norma entre dos vectores, el centroide o centro de la función y el punto del espacio multidimensional donde deseamos calcular el valor de la función. Si la red neuronal está formada por funciones de base radial la operación de interpolación de la curva de datos se expresa como [Powell, 1988]:

$$F(\mathbf{x}) = \sum_{i=1}^N \omega_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (3.2)$$

donde $\phi(\cdot)$ es la función de base radial, $\|\cdot\|$ es una norma determinada, N es el número de datos de entrenamiento, y \mathbf{x}_i es el centro de la función de base radial i -ésima. Los centros son los datos de entrada de entrenamiento. Por lo tanto el número de centros es N . La función F que genera la RBFNN debe cumplir la condición de interpolación $F(\mathbf{x}_i) = y_i$, donde y_i es el valor deseado de salida del conjunto de entrenamiento. La operación efectuada por la red neuronal se puede expresar de forma matricial:

$$\mathbf{y} = \Phi \boldsymbol{\omega} \quad (3.3)$$

donde $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ es el vector de salida deseado, Φ es una matriz de tamaño $N \times N$ con elementos $\Phi = \{\phi_{ji} = \phi(\|\mathbf{x}_j - \mathbf{x}_i\|) \text{ con } (j, i) = 1, 2, \dots, N\}$ y donde finalmente $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_N]^T$ es el vector de pesos. De la expresión (3.3) se extrae el vector de pesos aplicando:

$$\boldsymbol{\omega} = \Phi^{-1} \mathbf{y} \quad (3.4)$$

donde Φ^{-1} es la inversa de la matriz Φ .

La razón para utilizar funciones de base radial en las neuronas en lugar de otro tipo de funciones es debida al teorema de Micchelli [Micchelli, 1986]. El teorema anterior indica que si las funciones usadas en (3.2) son de tipo radial entonces la matriz de interpolación Φ es no singular. Existen dis-

tintas funciones que cumplen con las condiciones expuestas en el teorema de Micchelli, como por ejemplo la funciones multi-cuadráticas o las funciones gaussianas. Estas funciones poseen un factor que controla la rapidez o lentitud de su decaimiento. En las funciones gaussianas el mencionado factor que gobierna la "suavidad" de la función es la varianza. Posteriormente se comprobará la gran incidencia que posee este parámetro en la capacidad de aproximación de las RBFNNs gaussianas. El empleo de las funciones de base radial permite la construcción de una superficie F que pasa por todos los puntos de entrenamiento. Sin embargo, la interpolación estricta producida por las ecuaciones (3.2) y (3.4) implica una pobre generalización, especialmente cuando el número de datos de entrenamiento es elevado. En este caso, aparecerá el problema de sobre-entrenamiento. La RBFNN no generará un dato de salida apropiado cuando se presente un nuevo dato de entrada. Además, debido a la presencia de ruido en los datos de entrenamiento o la ausencia de datos en algunas zonas del espacio de entrada y salida, la construcción de la superficie de interpolación es un problema mal condicionado. En [Poggio y Girosi, 1990] se aplica la teoría de regularización para solucionar el problema del mal condicionamiento en la matriz de interpolación. La red de regularización expuesta en [Poggio y Girosi, 1990] es un aproximador universal y la aproximación realizada por esta red es óptima. Debido a que el entrenamiento de la red de regularización consume muchos recursos de computación, en [Poggio y Girosi, 1990] se propuso una solución sub-óptima consistente en la utilización de un número de neuronas M menor que el número de datos de entrenamiento N . En la notación de las siguientes secciones se utiliza \mathbf{c}_i para designar al centro de la neurona i -ésima. Este cambio en la notación es causado por el hecho de que no todos los datos de entrenamiento \mathbf{x} son centros neuronales. Se deberá seleccionar los M centros que proporcionan el mejor rendimiento. El algoritmo de entrenamiento puede incluso generar centros neuronales diferentes de los datos de entrenamiento aunque estén calculados a partir del conjunto de entrenamiento. Por tanto, ahora la operación que realiza la RBFNN es:

$$F(\mathbf{x}) = \sum_{i=1}^M \omega_i \phi_i(\|\mathbf{x} - \mathbf{c}_i\|) \quad (3.5)$$

La ecuación anterior se puede expresar de forma matricial como:

$$\mathbf{y} = \mathbf{P}\boldsymbol{\omega} \quad (3.6)$$

donde la matriz P sustituye a la matriz Φ de la ecuación (3.3). Los elementos de \mathbf{P} toman la siguiente expresión, $\mathbf{P} = \{p_{ji} = \phi(\|\mathbf{x}_j - \mathbf{c}_i\|)\}$ donde ahora los índices varían como $i = 1, 2, \dots, M$ y $j = 1, 2, \dots, N$.

Una RBFNN constituye un aproximador universal sin necesidad de incorporar ningún mecanismo de regularización. En [Park y Sandberg, 1991] se demuestra que una RBFNN compuesta por una única capa de neuronas con un mismo tipo de funciones de base radial y un factor de control de la suavidad de la función común a todas las neuronas constituye un aproximador universal. En efecto, si la función ϕ común a todas las neuronas es integrable, con un valor de integración acotado y distinto de cero, entonces la familia de funciones formada por la función neuronal es densa en el espacio de funciones de potencia p -ésima integrable. La propiedad de densidad permite asegurar que para

cualquier función objetivo f del espacio de funciones de potencia p -ésima integrable existirá una red neuronal $F(\mathbf{x})$ construida con las funciones ϕ tal que:

$$\|f(\mathbf{x}) - F(\mathbf{x})\| < \varepsilon \quad \text{Para todo } \mathbf{x} \in \mathcal{R}^n \text{ y } \varepsilon > 0 \quad (3.7)$$

donde \mathcal{R}^n es el conjunto de los vectores reales de dimensión n . En los teoremas desarrollados en [Park y Sandberg, 1991] la simetría radial de las funciones neuronales tiene que estar referida a un tipo de norma definida en \mathcal{R}^n . De esta forma, si $\|\mathbf{x}\| = \|\mathbf{y}\|$ entonces $\phi(\|\mathbf{x}\|) = \phi(\|\mathbf{y}\|)$. La norma más utilizada es la $\|\cdot\|_2$ o norma Euclídea.

Los enunciados teóricos pronostican la existencia, para cualquier función del tipo descrito anteriormente, de una RBFNN que lleva a cabo la aproximación con la precisión deseada siempre y cuando los parámetros de la red neuronal sean seleccionados de forma apropiada. Sin embargo, al igual que los teoremas empleados para demostrar la propiedad de aproximación universal en los MLP, el conjunto de resultados teóricos de [Park y Sandberg, 1991] no ofrece indicios prácticos sobre cómo seleccionar los parámetros que componen la red neuronal. Si los parámetros neuronales son elegidos de forma incorrecta, la aproximación será imprecisa, aunque la RBFNN sea un aproximador universal. Así, la tarea del algoritmo de entrenamiento consistirá en el cálculo de los parámetros neuronales a partir de los datos de entrenamiento con el objetivo de alcanzar la precisión prefijada. Las funciones de Green del medio apantallado multicapa son funciones continuas y presentan las características impuestas por los teoremas de [Park y Sandberg, 1991] para posibilitar la aproximación por medio de RBFNN. Sin embargo, la obtención de los parámetros adecuados de las RBFNNs es una tarea complicada. Diversos factores como el comportamiento singular en la fuente o la presencia de frecuencias de resonancia que producen fuertes variaciones en las funciones de Green complican su aproximación por medio de RBFNNs u otras redes neuronales.

La estructura de una RBFNN es similar a la de una SVM. Como se observa en la Figura 3.5 una RBFNN está formada por dos capas, la capa oculta y la capa de salida. El nivel de entrada no se considera una capa de la red neuronal ya que únicamente se encarga de la adquisición de datos. En la capa oculta se añade a las salidas de las neuronas no lineales un término constante o bias (b) de valor unidad que es multiplicado por un peso. La adición de este término constante suele mejorar la capacidad de aproximación de la red neuronal [Haykin, 1999]. Existen diferentes tipos de funciones de base radial aptas para servir de neurona en la capa oculta. Podemos citar las siguientes funciones de base radial:

$$\phi(\|\mathbf{x} - \mathbf{c}_i\|) = (\|\mathbf{x} - \mathbf{c}_i\|^2 + c^2)^{\frac{1}{2}} \quad \text{Función multi-cuadrática con } c > 0 \quad (3.8)$$

$$\phi(\|\mathbf{x} - \mathbf{c}_i\|) = \frac{1}{(\|\mathbf{x} - \mathbf{c}_i\|^2 + c^2)^{\frac{1}{2}}} \quad \text{Función inversa multi-cuadrática con } c > 0 \quad (3.9)$$

$$\phi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right) \quad \text{Función gaussiana con } \sigma > 0 \quad (3.10)$$

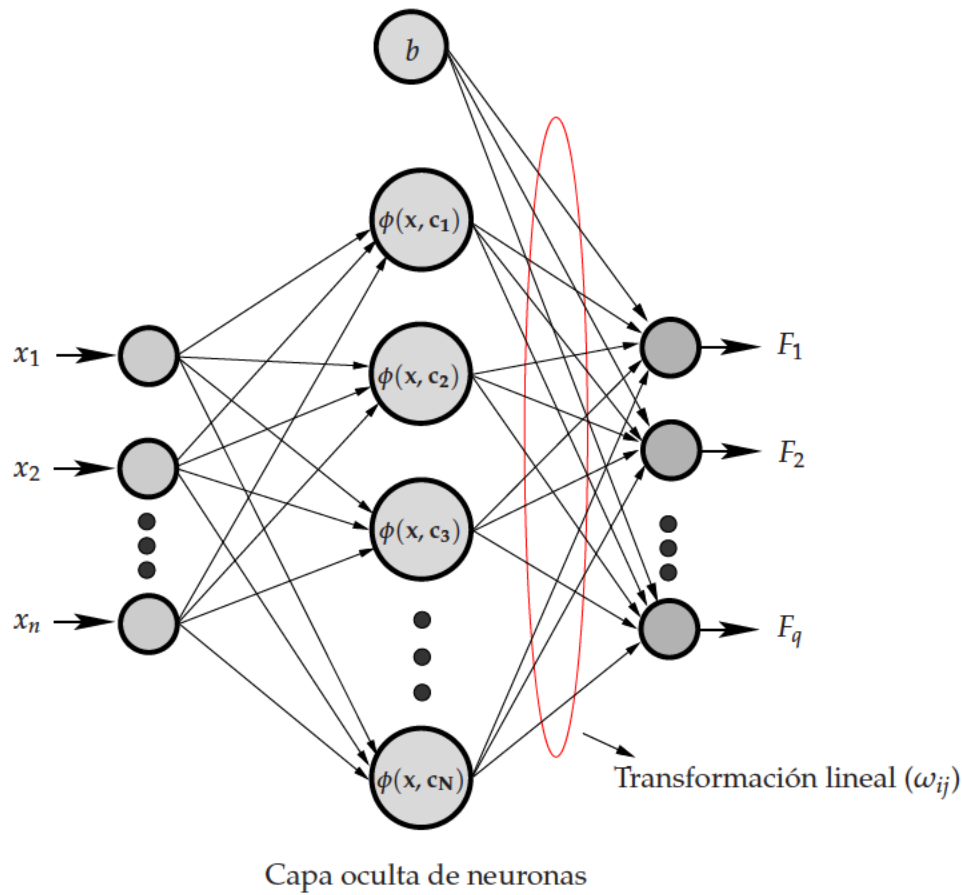


Figura 3.5: Estructura de una RBFNN. Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. La capa oculta realiza una transformación no lineal. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida. En cada función neuronal $\phi(x, c_i) = \phi(\|x - c_i\|)$. El término b es una constante que suele valer la unidad y está multiplicada por un conjunto de pesos adecuado.

La más utilizada de las funciones anteriores, tanto en problemas de clasificación y regresión, es la función gaussiana. En la ecuación (3.10), c_i es el centro de la función y σ_i^2 es la varianza de la gaussiana. Se puede fijar un valor de varianza común para todas las neuronas o por el contrario se puede asignar un valor de varianza particular para cada función gaussiana. Generalmente, el algoritmo de entrenamiento determina si la varianza es común o diferente en las neuronas gaussianas. Si los rangos de las coordenadas del espacio de entrada son diferentes puede ser apropiada la introducción de una norma ponderada. De esta forma cada una de las coordenadas tendrá una varianza propia [Poggio y Girosi, 1990]:

$$\phi(\|\mathbf{x} - \mathbf{c}_i\|) = e^{[-\frac{1}{2}(\mathbf{x} - \mathbf{c}_i)^T \Sigma^{-1}(\mathbf{x} - \mathbf{c}_i)]} \quad (3.11a)$$

$$\Sigma^{-1} = \mathbf{C}^T \mathbf{C} \quad (3.11b)$$

$$\mathbf{C} = \begin{pmatrix} 1/\sigma_1 & 0 & \cdots & 0 \\ 0 & 1/\sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\sigma_n \end{pmatrix} \quad (3.11c)$$

donde \mathbf{C} es la matriz de norma ponderada, Σ es la matriz de covarianza, n es la dimensión del espacio de entrada y \mathbf{c}_i el centro de la función gaussiana. Si la matriz de pesos es diagonal como la mostrada en la ecuación (3.11c) y además está compuesta por diferentes valores a lo largo de la diagonal, entonces cada coordenada del espacio de entrada tendrá una varianza particular. El uso de la matriz de covarianza y de norma ponderada puede ser evitado si los valores de cada coordenada son normalizados. En este caso existirá un rango de variación común a todas las coordenadas y se podrá aplicar un mismo valor de varianza a todas ellas $\sigma_1 = \sigma_2 = \dots = \sigma_i = \dots = \sigma_n$.

3.3. Algoritmos de entrenamiento de las Redes neuronales de funciones de base radial

El diseño de una RBFNN se desarrolla en dos fases: la fase de entrenamiento y la fase de validación. Durante la primera etapa, la RBFNN construye una superficie de interpolación a partir de un conjunto de pares de datos de entrada-salida. Posteriormente, en la fase de validación un nuevo conjunto de datos alimenta la red neuronal. El error cometido en la aproximación del nuevo conjunto de datos permite comprobar la capacidad de generalización de la red entrenada.

El proceso de entrenamiento consiste básicamente en la aplicación de un determinado algoritmo que calcule los parámetros neuronales. En el presente trabajo se ha optado por seleccionar como funciones neuronales las funciones gaussianas. El algoritmo de entrenamiento deberá seleccionar los centros de cada neurona así como los pesos de la capa de salida. La varianza será calculada bien por el algoritmo de entrenamiento o bien será definida por el diseñador de la RBFNN. Además, la varianza puede poseer un valor constante para todas las neuronas o por el contrario tomar un valor particular para cada neurona. En el caso de elegir una norma ponderada como la expuesta en la ecuación (3.11c) se deberá calcular o fijar las varianzas de cada coordenada. Existen dos estrategias para calcular los centros y, si es necesario, las varianzas neuronales. En las técnicas de entrenamiento supervisadas los valores de salida son tomados en cuenta para calcular los parámetros de las funciones gaussianas. En cambio, en las técnicas no supervisadas los centros y varianzas son estimados únicamente con los datos de entrada. Los pesos de la capa de salida siempre se calculan de forma supervisada. Esto es debido a que la composición lineal de los datos, proyectados en el espacio de la capa oculta, debe ajustarse de la mejor forma posible a los datos de salida. A continuación describiremos tres algoritmos de entrenamiento que han sido empleados en la aproximación de las funciones de Green de los potenciales del medio.

3.3.1. Selección de centros aleatoria

Esta estrategia de entrenamiento consiste en la selección aleatoria de los centros neuronales a partir del conjunto de datos de entrenamiento. Es por tanto una técnica no supervisada. Cada centro c_k es igual a un dato de entrada del conjunto de entrenamiento. La varianza de cada función gaussiana puede ser calculada como [Haykin, 1999]:

$$\sigma = \frac{d_{max}^2}{2m} \quad (3.12)$$

donde d_{max} es la máxima distancia entre los centros seleccionados y m es el número de centros. La elección de este valor de varianza evita la aparición de funciones gaussianas muy puntiagudas o muy planas. Los valores de varianza extremos pueden producir un rendimiento pobre. Aunque la varianza de la ecuación (3.12) proporciona generalmente un rendimiento adecuado, existen otras técnicas que estiman la varianza de una manera elaborada. En [Schwenker et al., 2001] se desarrollan varias expresiones de cálculo de la varianza que conducen a una mejor estimación del grado de variación de los datos de entrada. En cuanto a los pesos de salida se debe recurrir a una técnica como la de la matriz pseudo-inversa ya que la matriz \mathbf{P} de la ecuación (3.6) no es cuadrada. La operación llevada a cabo por la RBFNN se puede expresar de forma matricial como:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \varphi(\mathbf{x}_1, \mathbf{c}_1) & \varphi(\mathbf{x}_1, \mathbf{c}_2) & \cdots & \varphi(\mathbf{x}_1, \mathbf{c}_M) \\ \varphi(\mathbf{x}_2, \mathbf{c}_1) & \varphi(\mathbf{x}_2, \mathbf{c}_2) & \cdots & \varphi(\mathbf{x}_2, \mathbf{c}_M) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{x}_N, \mathbf{c}_1) & \varphi(\mathbf{x}_N, \mathbf{c}_2) & \cdots & \varphi(\mathbf{x}_N, \mathbf{c}_M) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_M \end{pmatrix} \quad (3.13)$$

$$\mathbf{y} = \mathbf{P}\boldsymbol{\omega} \quad (3.14)$$

donde \mathbf{y} es el vector de salidas de entrenamiento, $\boldsymbol{\omega}$ es el vector de pesos y \mathbf{P} es la matriz cuyos elementos son las funciones gaussianas evaluadas en los puntos de entrenamiento. Los pesos son calculados como:

$$\mathbf{w} = \mathbf{P}^+ \mathbf{y} \quad (3.15)$$

$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \quad (3.16)$$

donde \mathbf{P}^+ es la matriz pseudo-inversa de la matriz \mathbf{P} . Si el conjunto de entrenamiento es numeroso, el cálculo de la matriz pseudo-inversa es muy costoso. Para evitar este inconveniente existen distintos algoritmos eficientes de cómputo de la matriz pseudo-inversa [Golub y Loan, 1996].

3.3.2. Algoritmo de agrupación en K-medias (K-means)

La técnica de selección aleatoria de centros constituye un método de entrenamiento sencillo y rápido pero usualmente conduce a un rendimiento bajo. La aproximación es poco precisa o se generan RBFNNs más grandes de lo necesario. Además, si los centros resultan elegidos demasiado cerca unos de otros se pueden producir problemas de mal condicionamiento en la inversión de la matriz \mathbf{P} de la ecuación (3.6). El algoritmo de agrupación en K-medias (conocido también como algoritmo "K-means") intenta solucionar los problemas anteriores mediante la colocación de los centros de las gaussianas en aquellas zonas en las que existe una cantidad de datos relevante. Este algoritmo es especialmente útil cuando los datos se encuentran agrupados en diferentes núcleos. Como se observa en la Figura 3.6 estos núcleos pueden estar esparcidos en el espacio de entrada. El algoritmo de agrupación intenta distribuir los centros de manera equilibrada entre los diferentes núcleos de forma que la función objetivo quede representada correctamente.

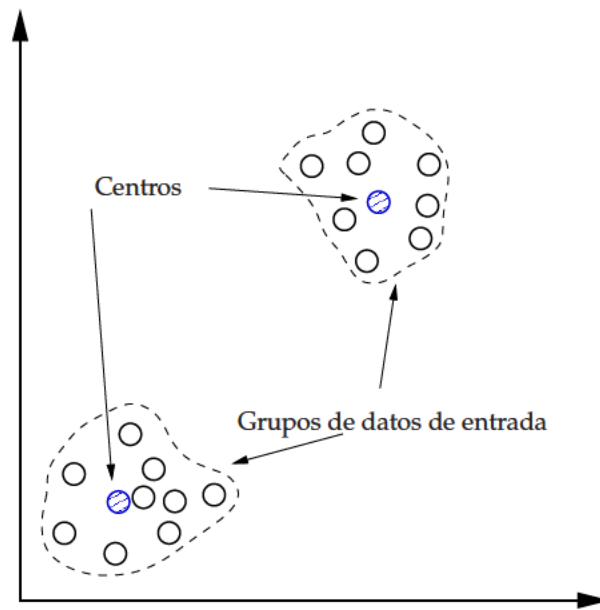


Figura 3.6: Grupos de datos de entrada del conjunto de entrenamiento y centros seleccionados por el algoritmo de agrupación de K-medias.

El algoritmo comienza con la selección aleatoria de M centros a lo largo del conjunto de entrenamiento. En cada iteración del algoritmo, un dato de entrada \mathbf{x} es asignado al grupo k -ésimo si se cumple la condición $\|\mathbf{x} - \mathbf{c}_k\| < \|\mathbf{x} - \mathbf{c}_j\|$ para $j = 1, 2, \dots, M$ y $j \neq k$. El nuevo centro se calcula como:

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{\mathbf{x} \in S_k} \mathbf{x} \quad (3.17)$$

donde N_k es el número de datos de entrada que pertenecen al grupo S_k . El grupo S_k se define como $S_k = \{\mathbf{x}, \|\mathbf{x} - \mathbf{c}_k\| < \|\mathbf{x} - \mathbf{c}_j\| \text{ para } j = 1, 2, \dots, M \text{ y } j \neq k\}$. Como se aprecia en la ecuación (3.17) los centros se calculan de forma no supervisada a partir de los datos de entrada del conjunto de entrenamiento. Los centros generados por este algoritmo no son iguales a ningún dato de entrada aunque se sitúen en las zonas del espacio donde existen datos de entrada (ver Figura 3.6). El proce-

dimiento se detiene cuando no existe variación en la posición de los centros o cuando se alcanza un número máximo de iteraciones.

Existen diversas formas de calcular la varianza de las funciones gaussianas. Generalmente, la varianza de cada función gaussiana es la media de las distancias entre un grupo de datos de entrada o entre centros vecinos. Por ejemplo, se puede seleccionar como valor de varianza de la gaussiana k -ésima, la media de las distancias a los p centros más cercanos [Schwenker et al., 2001]. Dicha media se pondera con un factor α de modo que obtenemos:

$$\sigma_k^2 = \left(\alpha \frac{1}{p} \sum_{i=1}^p d_i \right)^2 \quad (3.18)$$

donde d_i es la distancia entre el centro k -ésimo y el i -ésimo centro más cercano y $d_1 \leq \dots \leq d_i \leq \dots \leq d_M$. Usualmente, se emplea la norma Euclídea para calcular las distancias entre centros y datos de entrada. Otra alternativa para estimar σ^2 consiste en calcular la distancia entre el centro actual y el centro más cercano que pertenezca a otro grupo estableciendo la varianza como $\sigma = \alpha \min \{ \|c_j - c_k\| \}$ con $j = 1, 2, \dots, M$ y $j \neq k$. También se puede calcular una varianza para cada grupo con las distancias de los datos de entrada que pertenecen al grupo correspondiente.

3.3.3. Algoritmo de Mínimos cuadrados ortogonales (OLS)

El rendimiento de la RBFNN depende en gran medida de la adecuada selección de los centros de las neuronas. Las técnicas no supervisadas como la selección de centros aleatoria y la técnica de K-medias no proporcionan las RBFNNs más compactas debido a que únicamente hacen uso de los datos de entrada. Las técnicas no supervisadas colocan los centros en aquellas zonas del espacio de entrada donde existen datos de entrenamiento pero sin tener en cuenta en qué medida contribuyen a la representación de la función objetivo. Es posible que en una zona del espacio sean necesarios más centros que en otra zona, donde el comportamiento de la función es más predecible.

En el espacio de entrada existen una serie de puntos que incorporados como centros de las neuronas logran una correcta representación de la función objetivo. Para seleccionar dichos puntos del espacio se puede generar una función de error entre la función objetivo y la salida de la red neuronal. De esta forma, el algoritmo de entrenamiento supervisado buscará aquellos puntos que produzcan el mayor descenso en los niveles de la función de error. Por ello, las técnicas supervisadas alcanzan menores niveles de error para un mismo número de neuronas o bien un tamaño neuronal más reducido para un mismo nivel de error máximo respecto a los métodos no supervisados.

El algoritmo de Mínimos cuadrados ortogonales (OLS) es una técnica supervisada basada en la ortogonalización de la matriz \mathbf{P} de la ecuación (3.6) [Chen et al., 1991]. Para explicar el método de entrenamiento OLS empezaremos por el caso unidimensional, de modo que el espacio de salida es el espacio \mathbb{R} de los números reales. Teniendo en cuenta los errores de aproximación, la operación realizada por la RBFNN se expresa en el siguiente sistema matricial como:

$$\mathbf{y} = \mathbf{P}\boldsymbol{\omega} + \mathbf{E} \quad (3.19)$$

donde:

$$\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_N]^T \quad (3.20)$$

$$\mathbf{P} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \ \mathbf{p}_{M_s}] \quad (3.21)$$

$$\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \cdots \ \omega_{M_s}]^T \quad (3.22)$$

$$\mathbf{E} = [\epsilon_1 \ \epsilon_2 \ \cdots \ \epsilon_N]^T \quad (3.23)$$

Al igual que en la ecuación (3.6) \mathbf{y} es un vector columna que contiene los valores objetivo y $\boldsymbol{\omega}$ es un vector columna con los pesos lineales. El nuevo vector columna \mathbf{E} es la diferencia entre la estimación de la RBFNN y los valores objetivo, es decir, es un vector que contiene los errores que comete la red neuronal. La matriz \mathbf{P} está formada por una serie de vectores columna \mathbf{p}_i con $1 \leq i \leq M_s$. Los elementos de cada vector columna son fruto de evaluar la función ϕ , propia de las neuronas, en los puntos del conjunto de entrenamiento para un centro determinado (\mathbf{c}_i) de modo que cada elemento de \mathbf{p}_i es igual a $p_{ji} = \phi(\|\mathbf{x}_j - \mathbf{c}_i\|)$ con $1 \leq j \leq N$. Como se observa en la ecuación (3.21) existen M_s vectores columna obtenidos a partir de un número igual de centros elegidos entre los N datos de entrenamiento. La selección del subconjunto de M_s centros puede ser aleatoria. Generalmente no se descarta ningún dato de entrenamiento como centro de forma que $M_s = N$. A cada vector columna de \mathbf{P} se le denomina regresor:

$$\mathbf{p}_i = [p_{1i} \ p_{2i} \ \cdots \ p_{Ni}]^T \quad \text{regresor } i\text{-ésimo} \quad (3.24)$$

Los M_s regresores forman una base de vectores que generan un espacio. La solución mínimo cuadrática del sistema 3.19 garantiza que la solución encontrada ($\mathbf{P}\boldsymbol{\omega}$) es la proyección del vector \mathbf{y} en el espacio generado por los regresores. El cuadrado de la proyección de \mathbf{y} es parte de la energía de la solución deseada. Debido a que existe un cierto grado de correlación entre los distintos regresores, no es posible decidir cuánto contribuye un regresor individualmente a la energía de la función objetivo. Para seleccionar la mejor combinación de regresores, al existir correlación entre ellos, habría que probar todas las posibles combinaciones, de forma que el entrenamiento así realizado resultaría imposible. El método OLS transforma el conjunto de regresores p_i en un conjunto o base de vectores ortogonales de forma que se puede estimar la contribución a la energía de salida de cada uno de los vectores ortogonales de forma individual. El algoritmo proporciona así un modo de seleccionar paso a paso solo aquellos regresores (centros) que más aporten a la representación de la función objetivo. En consecuencia, es previsible que el número final de neuronas sea reducido, consiguiendo al mismo tiempo un bajo nivel de error en la aproximación de la función objetivo. En [Sherstinsky y Picard, 1996] se demuestra que el algoritmo OLS no es capaz de proporcionar el conjunto de regresores mínimo para alcanzar un determinado nivel de precisión en una tarea de aproximación. No obstante, a pesar de no garantizar la solución óptima, este algoritmo supervisado genera RBFNNs apropiadas para todo tipo de aplicaciones. El carácter supervisado le permite alcanzar bajos niveles de error con tamaños neuronales que aun siendo sub-óptimos no son muy grandes. Hay que tener en cuenta que para un conjunto de entrenamiento de N muestras existen un total de 2^N combinaciones diferentes de neuronas por lo que siempre es necesario emplear un algoritmo de

entrenamiento. Por tanto, en el entrenamiento de una RBFNN el algoritmo OLS es una alternativa válida.

El conjunto inicial de regresores puede ser producido a partir de un subconjunto de los datos de entrenamiento ($M_s < N$) o de forma más general a partir de todo el conjunto de entrenamiento ($M_s = N$). A partir del conjunto inicial de M_s regresores el algoritmo seleccionará aquellos M regresores que permiten alcanzar cierto nivel de error en la aproximación.

El algoritmo se fundamenta en la descomposición de la matriz \mathbf{P} en el producto de dos matrices:

$$\mathbf{P} = \mathbf{M}\mathbf{A} \quad (3.25)$$

donde \mathbf{A} es una matriz triangular de tamaño $M_s \times M$ con 1's en la diagonal y \mathbf{M} es una matriz con columnas ortogonales:

$$\mathbf{A} = \begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1M_s} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2M_s} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & \alpha_{M_s-1M_s} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \quad (3.26)$$

$$\mathbf{M} = [\mathbf{m}_1 \ \mathbf{m}_2 \ \cdots \ \mathbf{m}_{M_s}] \quad \mathbf{m}_i = [m_{1i} \ m_{2i} \ \cdots \ m_{Ni}]^T \quad 1 \leq i \leq M_s \quad (3.27)$$

Si sustituimos la ecuación (3.25) en la ecuación (3.19) obtenemos:

$$\mathbf{y} = \mathbf{M}\mathbf{A}\boldsymbol{\omega} + \mathbf{E} \quad (3.28)$$

Si al producto de la matriz \mathbf{A} por el vector de pesos $\boldsymbol{\omega}$ se le denomina vector \mathbf{g} podemos expresar la ecuación anterior como:

$$\mathbf{y} = \mathbf{M}\mathbf{g} + \mathbf{E} \quad (3.29)$$

Debido a que la matriz \mathbf{M} está compuesta por columnas ortogonales la solución mínimo cuadrática, $\hat{\mathbf{g}}$, de la ecuación (3.29) toma la expresión:

$$\mathbf{M}^T \mathbf{y} = \mathbf{M}^T \mathbf{M} \hat{\mathbf{g}} = \mathbf{H} \hat{\mathbf{g}} \quad (3.30)$$

Por tanto el vector $\hat{\mathbf{g}}$ se despeja como:

$$\hat{\mathbf{g}} = \mathbf{H}^{-1} \mathbf{M}^T \mathbf{y} \quad (3.31)$$

donde la matriz diagonal \mathbf{H} está formada por elementos de valor $h_i = \mathbf{m}_i^T \mathbf{m}_i$; $1 \leq i \leq M_s$ en la diagonal y 0's en el resto de posiciones. Su inversa es:

$$\mathbf{H}^{-1} = \begin{pmatrix} 1/h_1 & 0 & \cdots & 0 \\ 0 & 1/h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/h_{M_s} \end{pmatrix} \quad (3.32)$$

Por lo tanto cada elemento del vector $\hat{\mathbf{g}}$ se expresa como:

$$\hat{g}_i = \frac{\mathbf{m}_i^T \mathbf{y}}{h_i} = \frac{\mathbf{m}_i^T \mathbf{y}}{\mathbf{m}_i^T \mathbf{m}_i} \quad (3.33)$$

Los pesos se calculan a partir del sistema triangular:

$$\mathbf{A} \hat{\boldsymbol{\omega}} = \hat{\mathbf{g}} \quad (3.34)$$

El sistema anterior y el conjunto de vectores ortogonales de (3.27) se puede obtener mediante el empleo del método de Gram-Schmidt. Este procedimiento calcula en cada iteración una columna de la matriz \mathbf{A} y convierte en ortogonal un vector de \mathbf{P} respecto al conjunto de vectores ya ortogonalizados. El método OLS aplicado a la selección de un conjunto de regresores hace uso del algoritmo de Gram-Schmidt para incorporar a la base ortogonal aquel candidato que más aporte a la energía de salida. En la iteración k -ésima se calcula el resultado de ortogonalizar cada uno de los regresores respecto de los $k - 1$ vectores ya ortogonalizados. Aquel regresor que una vez ortogonalizado contribuya en mayor medida a la energía de la función objetivo será seleccionado. La contribución a la energía de salida se calcula con el candidato ortogonalizado ya que de este modo se encuentra incorrelado respecto a los regresores ya seleccionados. Así, su aportación a la función objetivo se evalúa de forma clara.

Para estimar la contribución de cada regresor ortogonalizado a la función o salida deseada comenzamos por definir la energía de la función objetivo como $\mathbf{y}^T \mathbf{y}$. Sustituyendo la expresión de la función objetivo de la ecuación (3.28) en la expresión de la energía de salida obtenemos:

$$\mathbf{y}^T \mathbf{y} = \hat{\mathbf{g}}^T \mathbf{M}^T \mathbf{M} \hat{\mathbf{g}} + \mathbf{E}^T \mathbf{E} \quad (3.35)$$

Como las columnas \mathbf{m}_i de la matriz \mathbf{M} son ortogonales los productos $\mathbf{m}_i \mathbf{m}_j$ son iguales a cero cuando $i \neq j$. Por lo tanto si se introduce la ecuación (3.33) en la ecuación (3.35), la energía de la función objetivo de salida equivale a:

$$\mathbf{y}^T \mathbf{y} = \sum_{i=1}^M \hat{g}_i^2 \mathbf{m}_i^T \mathbf{m}_i + \mathbf{E}^T \mathbf{E} \quad (3.36)$$

El término $\mathbf{E}^T \mathbf{E}$ es el error en la aproximación, es decir, la parte de la energía de salida no expresada por los regresores. En cambio, el término $\sum_{i=1}^M \hat{g}_i^2 \mathbf{m}_i^T \mathbf{m}_i$ corresponde a la parte de energía expresada por los regresores ortogonalizados. Así, se procede a la definición un ratio de reducción del error en la aproximación para cada regresor ortogonalizado:

$$err_i = \frac{\hat{g}_i^2 \mathbf{m}_i^T \mathbf{m}_i}{\mathbf{y}^T \mathbf{y}} \quad \text{Para } 1 \leq i \leq M_s \quad (3.37)$$

Cuanto mayor sea err_i mayor será la contribución del regresor ortogonalizado a la expresión de la energía de la función objetivo. El espacio generado por los regresores ortogonalizados es el mismo que el espacio de los regresores originales; por tanto el proceso sirve para seleccionar los centros de las neuronas. El proceso de entrenamiento comienza con la selección del primer regresor respecto al que van a ser ortogonalizados todos los demás regresores seleccionados. Para simplificar la notación consideramos que $\mathbf{g} = \hat{\mathbf{g}}$:

En la primera iteración $k = 1$

Para $1 \leq i \leq M_s$ se calcula

$$\mathbf{m}_i^1 = \mathbf{p}_i \quad \text{vector ortogonal provisional} \quad (3.38)$$

$$g_i^1 = \frac{(\mathbf{m}_i^1)^T \mathbf{y}}{(\mathbf{m}_i^1)^T \mathbf{m}_i^1} \quad (3.39)$$

$$err_i^1 = \frac{(g_i^1)^2 (\mathbf{m}_i^1)^T \mathbf{m}_i^1}{\mathbf{y}^T \mathbf{y}} \quad (3.40)$$

El superíndice en las ecuaciones anteriores indica la iteración mientras que el subíndice indica el regresor (columna de la matriz \mathbf{M}) para el que se calcula el ratio de error. Una vez calculados los ratios de error de la primer iteración err_i^1 se busca el regresor que produce la máxima reducción:

$$err_{i_1}^1 = \max \{err_i^1, \quad 1 \leq i \leq M_s\} \quad (3.41)$$

Así se selecciona:

$$\mathbf{m}_1 = \mathbf{m}_{i_1}^1 = \mathbf{p}_{i_1} \quad (3.42)$$

En la iteración k -ésima ($k \geq 2$) se habrán ortogonalizado $k - 1$ regresores y se tendrá que seleccionar el k -ésimo regresor para que sea incluido en la base de vectores ortogonales. El conjunto de regresores candidatos está compuesto por los regresores no ortogonalizados hasta la presente iteración, es decir, por $M_s - (k - 1)$ regresores. El regresor candidato seleccionado será aquel que,

una vez ortogonalizado, produzca la mayor reducción del error. Este parámetro de la ecuación (3.37) siempre se comprueba con el regresor una vez ortogonalizado ya que así se encuentra incorrelado con los regresores ortogonalizados. Por ello, la contribución del candidato ortogonalizado será independiente de la contribución de los regresores ya seleccionados y ortogonalizados. De este modo, la contribución del regresor candidato se podrá calcular de forma individual. A partir de la iteración $k \geq 2$ se deben calcular los valores α_{jk} de la matriz \mathbf{A} , ver ecuación (3.26), ya que son necesarios para ortogonalizar el regresor candidato respecto a los $(k-1)$ regresores previamente ortogonalizados. El subíndice k indica tanto la columna como la iteración. Dichos valores α_{jk} se calculan gracias a la expresión (3.25), así:

$$\mathbf{M}^T \mathbf{P} = \mathbf{M}^T \mathbf{M} \mathbf{A} = \mathbf{H} \mathbf{A} \quad \mathbf{A} = \mathbf{H}^{-1} \mathbf{M}^T \mathbf{P} \quad (3.43)$$

Debido a que se debe seleccionar el regresor que mayor reducción de error proporcione es necesario calcular una serie de coeficientes α_{jk}^i para cada uno de los regresores candidatos:

$$\alpha_{jk}^i = \frac{\mathbf{m}_j^T \mathbf{p}_i}{\mathbf{m}_j^T \mathbf{m}_j} \quad 1 \leq j \leq k-1 \quad (3.44)$$

En la ecuación 3.44 el regresor candidato \mathbf{p}_i no tiene superíndice ya que dicho regresor no depende de la iteración, su valor es el mismo para cualquier iteración. El superíndice i recorre los regresores desde $1 \leq i \leq M_s$ exceptuando aquellos regresores previamente seleccionados $i \neq i_1 \neq i_2 \neq \dots \neq i_{k-1}$. Una vez obtenidos los valores α_{jk}^i propios del regresor bajo estudio se calcula la reducción del error de la siguiente forma:

$$\mathbf{m}_i^k = \mathbf{p}_i - \sum_{j=1}^{k-1} \alpha_{jk}^i \mathbf{m}_j \text{ ortogonalización del candidato} \quad (3.45)$$

$$g_i^k = \frac{(\mathbf{m}_i^k)^T \mathbf{y}}{(\mathbf{m}_i^k)^T \mathbf{m}_i^k} \quad (3.46)$$

$$err_i^k = \frac{(g_i^k)^2 (\mathbf{m}_i^k)^T \mathbf{m}_i^k}{\mathbf{y}^T \mathbf{y}} \quad (3.47)$$

Al igual que en la primera iteración, se busca el regresor que produce la máxima reducción entre los regresores candidatos:

$$err_{i_k}^k = \max \{err_i^k, \quad 1 \leq i \leq M_s, \quad i \neq i_1 \neq i_2 \neq \dots \neq i_{k-1}\} \quad (3.48)$$

Hallado el valor máximo de error de la iteración k -ésima el nuevo vector ortogonal es:

$$\mathbf{m}_k = \mathbf{m}_{i_k}^k = \mathbf{p}_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk} \mathbf{m}_j \quad \text{con} \quad \alpha_{jk} = \alpha_{jk}^{i_k} \quad \text{y} \quad 1 \leq j < k \quad (3.49)$$

El algoritmo finaliza cuando se seleccionan M regresores (centros) del subconjunto inicial M_s de modo que $1 \leq k \leq M$.

El anterior algoritmo está diseñado para seleccionar un conjunto de regresores, o de forma equivalente centros de las neuronas, cuando la función de salida es unidimensional. Para conseguir una completa representación de las funciones de Green es necesario aproximar las partes reales e imaginarias de G_V, G_A^{xx}, G_A^{yy} . Por lo tanto, se debe emplear una versión del algoritmo OLS que calcule la reducción del error de los regresores en la aproximación de una función objetivo multi-dimensional [Chen et al., 1992]. Suponemos que la función de salida cuenta con q dimensiones:

$$\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \cdots \mathbf{y}_q] \quad \text{con} \quad \mathbf{y}_\eta = [y_{1\eta} \ y_{2\eta} \cdots y_{N\eta}]^T \quad \text{y} \quad 1 \leq \eta \leq q \quad (3.50)$$

En este caso los pesos de la capa de salida ω se agrupan en una matriz de tamaño $M_s \times q$. La columna η -ésima de la matriz de pesos contiene aquellos pesos que conectan las salidas de las neuronas con la coordenada correspondiente de la salida de la RBFNN. Asimismo, el vector \mathbf{g} del caso unidimensional se transforma en una matriz \mathbf{G} de tamaño $M_s \times q$:

$$\mathbf{Y} = \mathbf{P}\omega = \mathbf{M}\mathbf{G} \quad (3.51)$$

$$\omega = [\omega_1 \ \omega_2 \ \cdots \ \omega_q] \quad \text{donde} \quad \omega_\eta = [\omega_{1\eta} \ \omega_{2\eta} \ \cdots \ \omega_{M_s\eta}]^T \quad (3.52)$$

$$\mathbf{G} = [\mathbf{g}_1 \ \mathbf{g}_2 \ \cdots \ \mathbf{g}_q] \quad \text{donde} \quad \mathbf{g}_\eta = [g_{1\eta} \ g_{2\eta} \ \cdots \ g_{M_s\eta}]^T \quad (3.53)$$

$$\text{con} \quad 1 \leq \eta \leq q$$

Para calcular el ratio de reducción del error en la aproximación de la función objetivo es necesario estimar la energía de salida. Debido a que \mathbf{y} es una matriz, debemos recurrir a la matriz de covarianza $\mathbf{y}^T \mathbf{y}$. La energía de salida se estima como la traza de la mencionada matriz. Al igual que en el caso unidimensional (3.35), se puede conseguir una expresión en función de los vectores ortogonales y los coeficientes $g_{i\eta}$:

$$\text{Traza}(\mathbf{y}^T \mathbf{y}) = \sum_{\eta=1}^q \mathbf{y}_\eta^T \mathbf{y}_\eta + \text{Traza}(\mathbf{E}^T \mathbf{E}) = \sum_{\eta=1}^q \sum_{i=1}^M g_{i\eta}^2 (\mathbf{m}_i^T \mathbf{m}_i) + \text{Traza}(\mathbf{E}^T \mathbf{E}) \quad (3.54)$$

Del mismo modo que en el caso unidimensional, la energía de la función objetivo está compuesta por dos términos: la energía del error y la contribución de los vectores ortogonales. Así, es posible encontrar una expresión que indique la reducción del error en la aproximación para la iteración k -ésima:

$$err_i^k = \frac{\sum_{\eta=1}^q (g_{i\eta}^k)^2 ((\mathbf{m}_i^k)^T \mathbf{m}_i^k)}{\sum_{\eta=1}^q \mathbf{y}_\eta^T \mathbf{y}_\eta} \quad (3.55a)$$

$$\text{con } g_{i\eta}^k = \frac{(\mathbf{m}_i^k)^T \mathbf{y}_\eta}{(\mathbf{m}_i^k)^T \mathbf{m}_i^k} \quad (3.55b)$$

$$1 \leq i \leq M_s \text{ excepto } i \neq i_1 \neq i_2 \neq \dots \neq i_{k-1} \text{ y } 1 \leq k \leq M \ (M < M_s)$$

El algoritmo descrito para el caso unidimensional en las ecuaciones (3.45) y (3.49) se aplica del mismo modo al caso multi-dimensional. El único cambio radica en el uso de la ecuación de reducción del error (3.55a) en lugar de la ecuación (3.37).

El algoritmo OLS posibilita la selección supervisada de centros y, si se desea también, de los pesos de la capa de salida lineal. Dependiendo de la función de base radial empleada será necesario fijar ciertos parámetros previamente al comienzo del algoritmo. En el presente trabajo se han utilizado funciones gaussianas debido a sus buenas características en tareas de aproximación de funciones. Como ya se ha mencionado, las funciones gaussianas son caracterizadas por su centro y varianza. Por lo tanto, será necesario prefijar el valor de varianza antes del inicio del algoritmo de entrenamiento. Como se mostrará en la Sección 3.5 la varianza tiene una importancia vital en la capacidad de la RBFNN en aproximar una función objetivo. Si este parámetro se fija de forma adecuada el rendimiento del algoritmo OLS se verá incrementado ampliamente.

Existen otros algoritmos supervisados que evalúan los centros, pesos e incluso varianzas de una RBFNN. La definición de una función de error entre la salida de la RBFNN y el valor deseado permite la utilización de técnicas de optimización que busquen aquellos parámetros que minimizan la función de error. Existen diversos algoritmos que hacen uso del gradiente de la función de error durante el proceso de optimización [Haykin, 1999, Fernandez-Redondo et al., 2004]. La técnica de optimización de los Algoritmos genéticos también se ha aplicado al problema de la búsqueda de los parámetros de una RBFNN como en [Bai y Zhang, 2002]. Estas técnicas presentan inconvenientes que impiden su uso en la tarea de aproximación de las funciones de Green. Las técnicas basadas en el gradiente de la función de error necesitan la inicialización de los parámetros a calcular. Si el número de parámetros es relativamente elevado, la inicialización puede resultar decisiva en el rendimiento de la aproximación de la función objetivo. Por ejemplo en [Kampolis et al., 2004] se recurre al método OLS para realizar una preselección de centros entre todos los datos de entrenamiento. Además, si el tamaño del conjunto de entrenamiento es grande, es probable que el algoritmo de optimización basado en el gradiente quede atrapado en un mínimo local y no alcance el error mínimo global. En el caso de la aproximación de las funciones de Green se cumplen las condiciones que descartan a las técnicas basadas en el uso del gradiente. Para evitar los inconvenientes propios de las técnicas basadas en gradiente, se puede acudir a técnicas de optimización estocásticas como los mencionados Algoritmos genéticos. Estos métodos de diseño de una RBFNN presentan como principal inconveniente la lentitud en la computación de los parámetros necesarios. Para no convertir en poco práctico el método de análisis neuronal se descartaron estas técnicas.

3.4. Extracción de la singularidad

En la Sección 2.5.1 se mostró el comportamiento singular de las funciones de Green de los potenciales del medio multicapa apantallado. La singularidad provoca un fuerte crecimiento de las funciones de Green en las cercanías de la fuente (ver Figuras 2.14 y 2.15). Como se ha explicado en la Sub-sección 3.2.3, la operación que realiza una RBFNN, especialmente en una tarea de regresión, equivale a la construcción de una superficie que aproxima a una curva deseada. Cada nuevo dato de salida es el resultado de una interpolación multidimensional generada a partir de los datos con los que ha sido entrenada la red neuronal. Si la función objetivo presenta fuertes variaciones, los datos de salida correspondientes al conjunto de entrenamiento pueden diferir sustancialmente de los datos del conjunto de validación. En este caso, la RBFNN no será capaz de generalizar correctamente. Los datos de entrenamiento deben estar situados en el espacio de entrada adecuadamente, de forma que el comportamiento de la función objetivo quede reproducido fielmente. El crecimiento de las funciones de Green, conforme el punto observación se acerca a la fuente, es muy rápido, de modo que son necesarios una gran cantidad de puntos para expresar correctamente dicho crecimiento. En este caso, el tiempo de entrenamiento y el tamaño de las redes neuronales convertirían al método neuronal en poco provechoso.

Sin embargo, la singularidad presente en la fuente puede ser extraída de forma que el comportamiento de las funciones de Green sea mucho menos abrupto. El método empleado para calcular las funciones de Green divide el espacio en el que se sitúan los puntos observación en dos zonas. En la zona más lejana de la fuente o zona DS (débil singularidad) se utiliza una técnica de aceleración en la suma de la serie espectral. En la zona más cercana o zona MS (máxima singularidad) se aplica un método híbrido ya que la parte dinámica es sumada en el dominio espectral, mientras que la parte cuasi-estática es evaluada en el dominio espacial mediante una suma de imágenes espaciales. Debido a que el conjunto infinito de imágenes sustituye a las paredes metálicas de la estructura, las funciones de Green empleadas para calcular el valor de las imágenes son aquellas correspondientes al caso de espacio libre o medio multicapa sin paredes. Las expresiones de las mencionadas funciones de Green pueden observarse en las ecuaciones (2.35) y (2.40). Se aprecia que el comportamiento singular de las funciones de Green es producido por el término $(1/\rho)$ donde (ρ) es la distancia entre punto observación y el punto fuente. Así, la multiplicación de las funciones de Green por el término (ρ) será suficiente para eliminar la singularidad en las cercanías de la fuente. Por ello, las RBFNNs aproximarán las funciones de Green multiplicadas por (ρ) ($G_V \cdot \rho, G_A^{xx} \cdot \rho, G_A^{yy} \cdot \rho$). Para recuperar la aproximación del valor original en la evaluación de la IE, habrá que dividir el valor de salida de las RBFNNs por la distancia entre punto fuente y observación.

Para mostrar de forma práctica el cambio en la variación de las funciones de Green producida por la extracción de la singularidad recurrimos a una serie de simulaciones similares a la efectuadas en la Sección 2.5.1. En la presente Sección se ha empleado una cavidad diferente (ver Figura 3.7) a la utilizada en 2.5.1. Se muestra el comportamiento de las funciones de Green en una estructura y frecuencia de trabajo completamente diferentes para ampliar la visión que se tiene sobre el comportamiento de las funciones de Green. Además, en esta estructura se analizará un filtro paso bajo con el método neuronal. Al igual que en la Sección 2.5.1, los puntos observación llenan por completo la cavidad y se distribuyen en forma de rejilla regular.

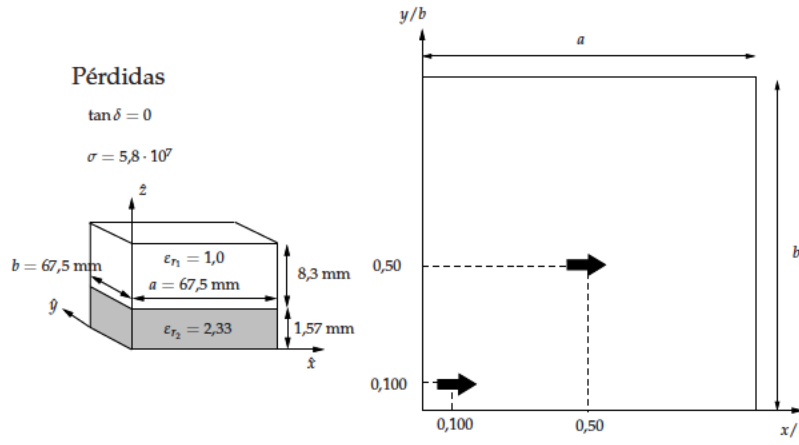


Figura 3.7: Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado en el centro del plano transversal.

En primer lugar estudiaremos las variaciones de las funciones de Green cuando la estructura es apantallada lateralmente de forma que tanto la tapa superior como inferior están abiertas. En las Figuras 3.8 y 3.9 podemos observar el rápido crecimiento de la función de Green del potencial eléctrico escalar para dos situaciones distintas. Por un lado para un dipolo situado en el centro de la cavidad y por otro lado para un dipolo ubicado cerca de las paredes. En ambos casos, la multiplicación por (ρ) permite obtener una función de variación mucho más lenta que la función original (Figuras 3.10 y 3.11). La eliminación de la singularidad provoca incluso un decaimiento en los valores de la función cerca de la fuente cuando el dipolo se encuentra en el centro del plano XY .

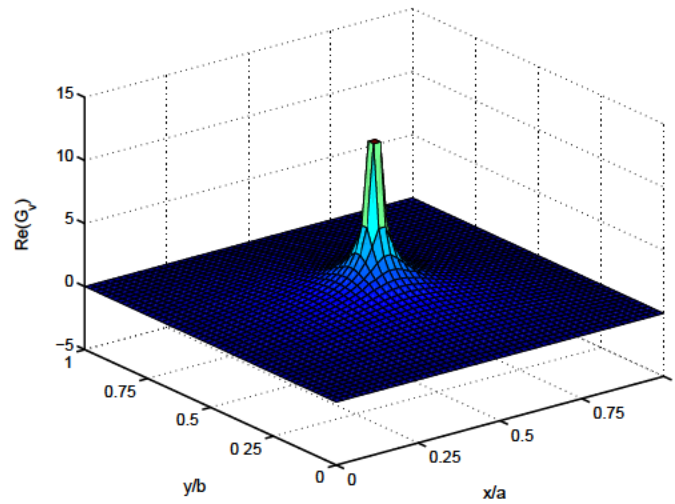


Figura 3.8: Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en el centro de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.

Cuando se colocan muros metálicos en las tapas superior e inferior, el crecimiento de la función de Green bajo estudio es en general más rápido, de manera que la dificultad de la tarea de aproximación aumenta. El carácter abrupto de la función de Green se produce tanto lejos como cerca de

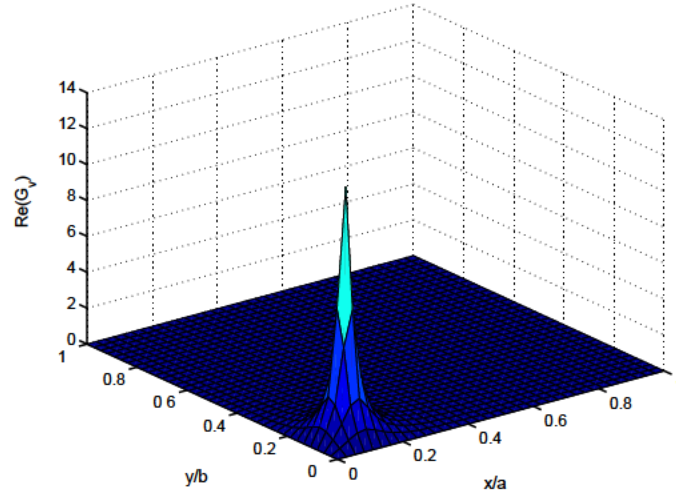


Figura 3.9: Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en cerca de las paredes de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.

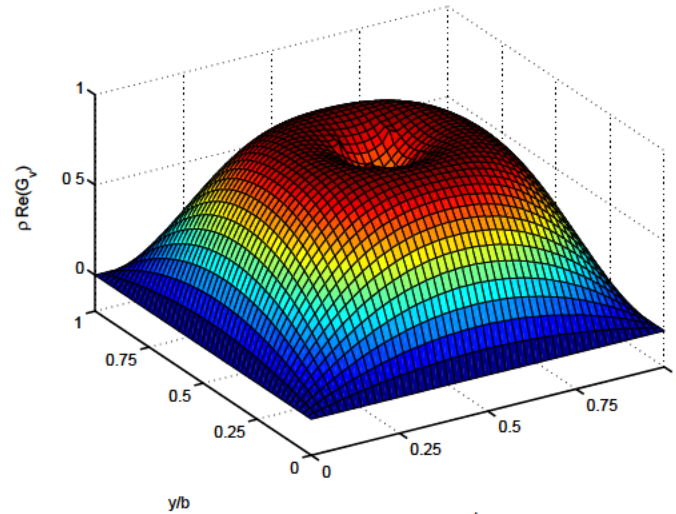


Figura 3.10: Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado en el centro de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.

las paredes tal como se observa en la Figuras 3.12 y 3.13. Por ello debemos multiplicar las funciones de Green por la distancia fuente-observación (ρ). En nuestro ejemplo se obtienen las funciones de las Figuras 3.14 y 3.15. Tanto si el dipolo se ubica en el centro de la cavidad como cerca de las paredes el crecimiento de la función es más suave. En el caso de no extraer la singularidad el valor de la función de Green tiende hacia infinito conforme nos acercamos a la fuente. Sin embargo, al multiplicar por la distancia (ρ), la función de Green tiende hacia un valor finito. Por tanto, la extracción de la singularidad facilita la aproximación por medio de RBFNNs tal y como se mostrará en las siguientes secciones.

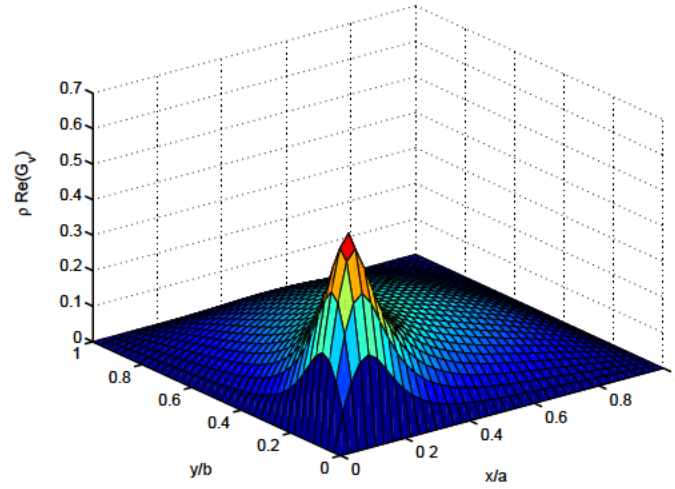


Figura 3.11: Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.

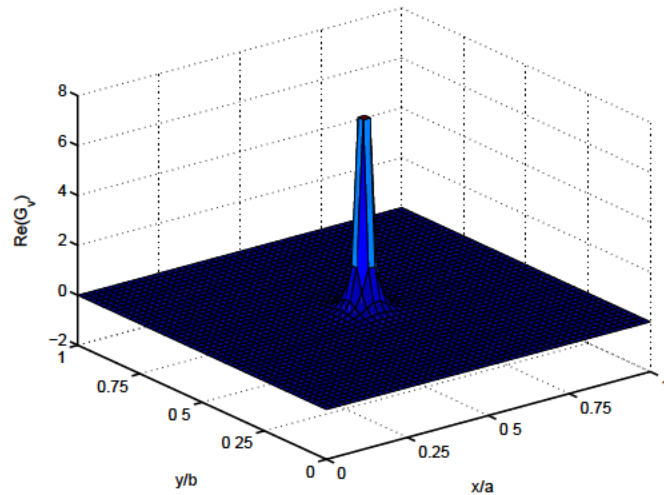


Figura 3.12: Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en el centro de la cavidad representada en la Figura 3.7.

3.5. Rendimiento de los algoritmos de entrenamiento en la aproximación de las funciones de Green

En este apartado se analiza el rendimiento de los tres algoritmos de entrenamiento de una RBFNN precedentes en el problema de aproximación de las funciones de Green en un ejemplo práctico. El objetivo del análisis es determinar qué algoritmo es el más apropiado para alcanzar la precisión requerida. El algoritmo más eficiente será aquel que requiera el menor número de neuronas para conseguir la mencionada precisión. Además, el tiempo invertido para generar las neuronas y los pesos deberá permitir la aplicación práctica del método neuronal. Con el objetivo de comprobar la habili-

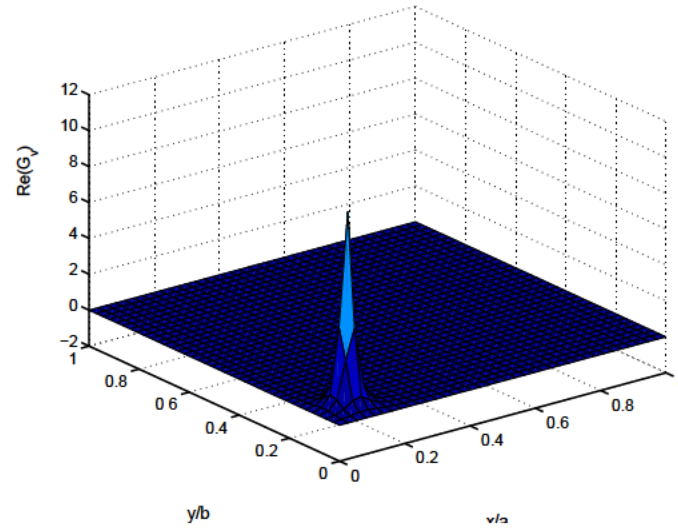


Figura 3.13: Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7.

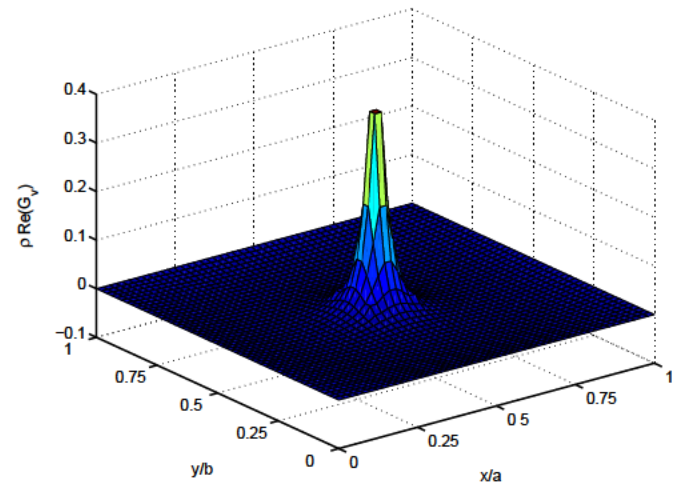


Figura 3.14: Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado en el centro de la cavidad representada en la Figura 3.7.

dad de los algoritmos de entrenamiento se ha generado una rejilla de puntos fuente que cubren el primer cuadrante de la estructura de la Figura 3.7. Para cada punto fuente se genera una rejilla regular de puntos observación que cubren toda la cavidad. De esta forma las coordenadas de los puntos fuente varían desde $(0 \leq x'/a \leq 0,5 ; 0 \leq y'/a \leq 0,5)$ mientras que los puntos observación poseen coordenadas un rango de variación desde $(0 \leq x/a \leq 1 ; 0 \leq y/a \leq 1)$. Debido a la propiedad de simetría expuesta en la Sección 2.5.2 las funciones de Green de cualquier pareja de puntos fuente-observación son iguales a las funciones de Green de un punto fuente simétrico situado en el primer cuadrante y un punto observación apropiado. Dependiendo de la posición del punto observación original será necesario realizar también una traslación de dicho punto para recuperar las funciones

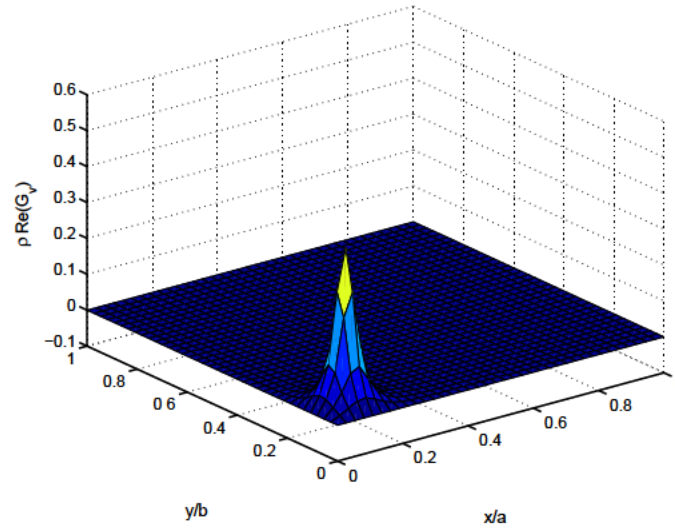


Figura 3.15: Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7.

de Green correctas (ver ecuaciones (2.78)). Por lo tanto, gracias a la propiedad de simetría, la rejilla de puntos fuente del conjunto de entrenamiento se limita al primer cuadrante. Durante la resolución de la IE existirán puntos fuente situados en cuadrantes distintos del primero. En este caso será necesario realizar las transformaciones oportunas mostradas en (2.78) para que la RBFNN ofrezca los valores correctos.

En la Figura 3.16 se muestra la disposición de los puntos fuente y observación tanto para el conjunto de entrenamiento como para el conjunto de validación o test. El esquema de generación de los mencionados conjuntos se ha aplicado a la estructura dibujada en la Figura 3.7. No obstante, es un esquema genérico que puede ser utilizado en cavidades rectangulares de diferentes dimensiones ya que las coordenadas espaciales están normalizadas. Como se observa en la Figura 3.16, los puntos observación del conjunto de entrenamiento se distribuyen por todo el plano transversal incluyendo las paredes metálicas. De esta forma, los datos de entrenamiento aportan la información necesaria sobre las condiciones de contorno en la cavidad. Los puntos fuente y observación del conjunto de validación se encuentran situados en posiciones intermedias respecto a los puntos de entrenamiento. Así, se comprueba de forma efectiva la capacidad de generalización de la RBFNN entrenada. Con el muestreo realizado en el espacio de entrada el conjunto de entrenamiento esta compuesto por un total de 8281 datos mientras que el de validación cuenta con 17316 muestras.

La RBFNN debe contar con seis salidas para aproximar completamente las funciones de Green del medio apantallado multicapa. Tres salidas corresponden a las partes reales de las funciones de Green y otras tres corresponden a las partes imaginarias. Como se indicó en la Sección 3.4 es necesario multiplicar las funciones de Green por la distancia (ρ) entre punto observación y fuente para eliminar la singularidad en la fuente. Gracias a esta operación el comportamiento de las funciones de Green es mucho menos abrupto, de forma que su aproximación por parte de una RBFNN se vuelve posible. El número de entradas de la RBFNN es de cuatro ya que cada dato de entrada está compuesto por las

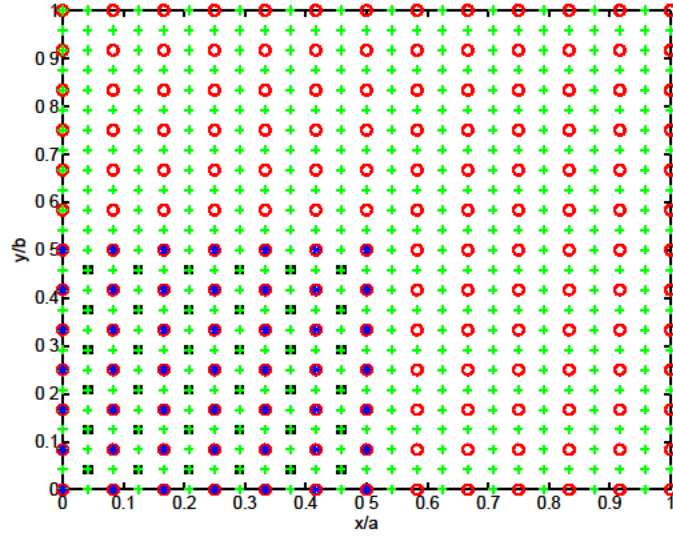


Figura 3.16: Rejilla de puntos de entrenamiento y validación. Los puntos fuente de entrenamiento son asteriscos y los puntos observación son círculos. Los puntos fuente del conjunto de validación se representan con cuadrados y los puntos observación con signos suma.

coordenadas espaciales normalizadas del punto fuente y del punto observación. Por tanto, la RBFNN va a realizar una proyección (mapeo) entre un espacio de entrada de dimensión cuatro y un espacio de salida de dimensión seis: $\mathbb{R}^4 \rightarrow \mathbb{R}^6$. La estructura de la RBFNN aplicada en la aproximación de las funciones de Green se observa en la Figura 3.17. La expresión matemática correspondiente a la operación de la RBFNN es:

$$F_{\eta}(\mathbf{x}) = b\omega_{0\eta} + \sum_{i=1}^M \omega_{i\eta} \exp \left[-\frac{1}{2}(\mathbf{x} - \mathbf{c}_i)^T \Sigma^{-1}(\mathbf{x} - \mathbf{c}_i) \right] \quad 1 \leq \eta \leq 6 \quad (3.56)$$

donde b es el término constante o de bias de valor igual a la unidad y \mathbf{c}_i el centro de la gaussiana.

Generalmente, el algoritmo requerirá el ajuste o definición de ciertos factores por parte del usuario. Una mala asignación en dichos factores puede provocar el funcionamiento incorrecto del algoritmo. Conforme el número de factores a ajustar de forma heurística se reduce, el algoritmo de entrenamiento será más versátil y resistente a errores por parte del diseñador. En el caso del algoritmo OLS la varianza es un parámetro que debe ser calculado o fijado previamente al inicio del entrenamiento. En las estrategias no supervisadas existen algoritmos que calculan un valor de varianza adecuado pero necesitan del ajuste de ciertos parámetros de forma heurística. Debido a que el rango de variación de cada una de las dimensiones del espacio de entrada es similar, se utiliza un valor de varianza común a todas ellas. En los algoritmos de selección de centros aleatorios y agrupación en K-medias se empleó la técnica de la media de las distancias a los p centros más cercanos para calcular las varianzas gaussianas [Schwenker et al., 2001]. Cada centro posee su propia varianza, que depende de las distancias a los centros más cercanos. Para realizar dicha media se eligieron en cada centro los 25 más cercanos y el parámetro α de la ecuación (3.18) se fijó tras realizar varias pruebas. En el mecanismo de agrupación de K-medias se realiza un máximo de 20 iteraciones del

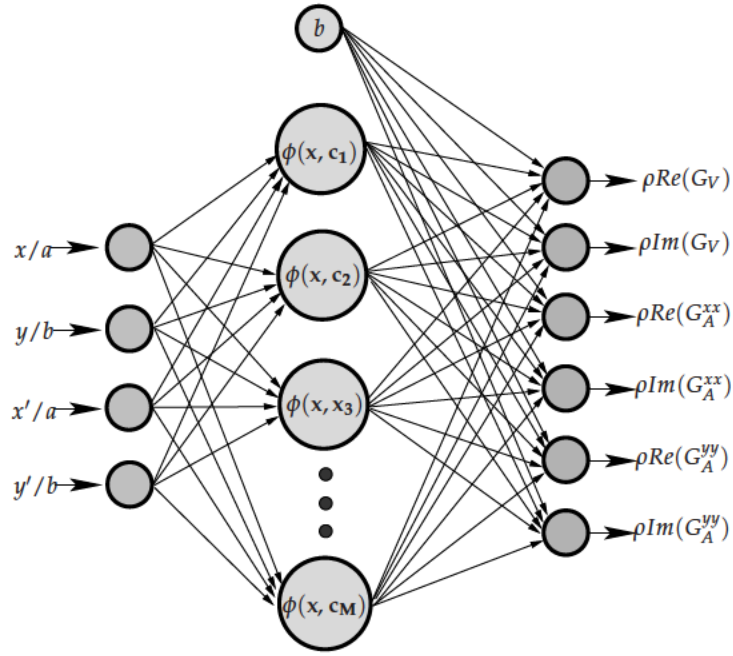


Figura 3.17: Estructura de la RBFNN que aproxima las partes reales e imaginarias de las funciones de Green multiplicadas por (ρ) .

algoritmo descrito en la Sección 3.3.2 para ajustar la posición de los centros y sus grupos correspondientes. El algoritmo OLS impide la estimación de las varianzas mediante técnicas como las descritas en [Schwenker et al., 2001]. Esto es debido a que la selección de los centros es supervisada y las varianzas influyen en dicha selección. Para cada situación estudiada se han probado diversos valores de varianza con el objetivo de analizar el rendimiento de este algoritmo en función de este parámetro neuronal.

De los tres algoritmos analizados, el OLS es aquel que más tiempo consume en el diseño de la RBFNN. Para reducir el periodo de entrenamiento invertido por este algoritmo se han realizado una serie de modificaciones y simplificaciones sobre las expresiones mostradas en la Sección 3.3.3. Según se explicó en dicha Sección, en la iteración k -ésima del algoritmo OLS se calculan los $k - 1$ coeficientes α_{jk}^i correspondientes al regresor i -ésimo. Como se indica en la ecuación (3.45) los coeficientes mencionados sirven para ortogonalizar el i -ésimo regresor candidato a ingresar en el conjunto de vectores ortogonales. La ecuación (3.45) implica que en cada iteración se realiza la resta entre el regresor i -ésimo y los $k - 1$ vectores ortogonales multiplicados por los coeficientes α_{jk}^i . De esta forma, al regresor \mathbf{p}_i se le resta un primer término $\alpha_{1k}^i \mathbf{m}_1$, un segundo término $\alpha_{2k}^i \mathbf{m}_2$, hasta el último término $\alpha_{(k-1)k}^i \mathbf{m}_{k-1}$. Si el regresor i -ésimo no es seleccionado en la iteración k entonces el mismo procedimiento es repetido en la siguiente iteración con un nuevo término en la resta igual a $\alpha_{k(k+1)}^i \mathbf{m}_k$. Excepto este último término todos los demás términos son iguales a los calculados en la iteración k -ésima, debido a que los coeficientes α_{jk}^i no dependen del índice k como se puede apreciar en la ecuación (3.44). Así, en lugar de calcular los coeficientes α_{jk}^i y la resta de términos en cada iteración k , existe un modo más eficiente de realizar la operación de ortogonalización del regresor candidato. En efecto, la resta del término k -ésimo de la ecuación (3.45) se puede realizar al final de la selección del regresor ortogonalizado en la correspondiente iteración número k . De este modo, al final de la

iteración k -ésima se obtienen los candidatos m_i^{k+1} necesarios en la siguiente iteración.

En la primera iteración, el algoritmo opera según lo indicado en la ecuaciones (3.38-3.40). Una vez seleccionado el primer regresor ortogonal, se añaden la siguiente serie de operaciones que impliquen la resta del primer término de la ecuación (3.45) a cada uno de los regresores no seleccionados:

$$\mathbf{m}_i^2 = \mathbf{p}_i - \alpha_{12}^i \mathbf{m}_1 = \mathbf{p}_i - \frac{\mathbf{m}_1^T \mathbf{p}_i}{\mathbf{m}_1^T \mathbf{m}_1} \mathbf{m}_1 \quad \text{para } 1 \leq i \leq M_s \quad i \neq i_1 \quad (3.57)$$

Hay que tener en cuenta que el vector \mathbf{m}_1 es el primer regresor seleccionado y por tanto el vector respecto al que se va a ortogonalizar el segundo regresor seleccionado. En cambio, los vectores \mathbf{m}_i^2 son regresores candidatos a ser ortogonalizados en la segunda iteración. Estos vectores pueden ser utilizados directamente en la ecuación (3.46) de cálculo de los coeficientes g_i^k ($g_{i\eta}^k$ del caso multidimensional). Estos coeficientes se emplean para calcular el ratio de reducción del error. Recordemos que el regresor que consiga la máxima reducción resulta seleccionado. Una vez seleccionado dicho regresor, toma el nombre de $\mathbf{m}_2 = \mathbf{m}_{i_2}^2$ y es empleado para generar el segundo término de la resta (3.45). Dicho término permite calcular los vectores candidatos \mathbf{m}_i^3 de la tercera iteración:

$$\mathbf{m}_i^3 = \mathbf{m}_i^2 - \alpha_{23}^i \mathbf{m}_2 = \mathbf{m}_i^2 - \frac{\mathbf{m}_2^T \mathbf{m}_i^2}{\mathbf{m}_2^T \mathbf{m}_2} \mathbf{m}_2 \quad \text{para } 1 \leq i \leq M_s \quad i \neq i_1 \neq i_2 \quad (3.58)$$

En la iteración k -ésima tenemos que seleccionar el vector $\mathbf{m}_k = \mathbf{m}_{i_k}^k$:

$$\mathbf{m}_i^{k+1} = \mathbf{m}_i^k - \alpha_{k+1}^i \mathbf{m}_k = \mathbf{m}_i^k - \frac{\mathbf{m}_k^T \mathbf{m}_i^k}{\mathbf{m}_k^T \mathbf{m}_k} \mathbf{m}_k \quad \text{para } 1 \leq i \leq M_s \quad i \neq i_1 \neq i_2 \cdots \neq i_k \quad (3.59)$$

El procedimiento sigue sucesivamente hasta que el último regresor M -ésimo es elegido ($1 \leq k \leq M$). Esta transformación se aplica de igual forma al caso multidimensional. Si se desea que el algoritmo OLS calcule también los pesos de la capa de salida, los coeficientes, α_{jk} , $1 \leq j < k$ de la iteración k -ésima se pueden calcular con el regresor seleccionado $m_k^{i_k}$.

Una segunda simplificación permite una disminución considerable del número de operaciones invertidas. En efecto, si sustituimos la ecuación (3.46) en el ratio de reducción del error (3.47) obtenemos:

$$err_i^k = \frac{[(\mathbf{m}_i^k)^T \mathbf{y}] / [(\mathbf{m}_i^k)^T \mathbf{m}_i^k]}{\mathbf{y}^T \mathbf{y}} = \frac{((\mathbf{m}_i^k)^T \mathbf{y})^2}{((\mathbf{m}_i^k)^T \mathbf{m}_i^k) \mathbf{y}^T \mathbf{y}} \quad (3.60)$$

Asimismo, en el caso multidimensional se puede sustituir los coeficientes $g_{i\eta}^k$ en la ecuación del ratio de reducción del error (3.55a). Así, se consigue una simplificación similar a la obtenida en el caso unidimensional:

$$err_i^k = \frac{\sum_{\eta=1}^q [((\mathbf{m}_i^k)^T \mathbf{y}_j) / ((\mathbf{m}_i^k)^T \mathbf{m}_i^k)]^2 (\mathbf{m}_i^k)^T \mathbf{m}_i^k}{\sum_{\eta=1}^q \mathbf{y}_j^T \mathbf{y}_j} = \frac{\sum_{\eta=1}^q ((\mathbf{m}_i^k)^T \mathbf{y}_j)^2}{((\mathbf{m}_i^k)^T \mathbf{m}_i^k) \sum_{\eta=1}^q \mathbf{y}_j^T \mathbf{y}_j} \quad (3.61)$$

Las simplificaciones anteriores permiten utilizar en las ecuaciones de reducción del error los vectores candidatos \mathbf{m}_i^k . Por tanto, no es necesario calcular los coeficientes g_i^k en el caso unidimensional ni $g_{i\eta}^k$ en el multidimensional durante el cálculo de la reducción del error. Además hay que tener en cuenta que el cálculo de los coeficientes de reducción del error solo se realiza para aquellos regresores no seleccionados. Una vez seleccionado el vector \mathbf{m}_i^k que debe ser incluido en la base ortogonal, se pueden calcular los coeficientes g definitivos si se desea que el algoritmo OLS calcule también los pesos de la capa lineal.

Para medir las diferencias entre los valores de salida de la RBFNN respecto a los valores deseados de las funciones de Green se utiliza el error cuadrático medio normalizado (“Normalized mean squared error” o NMSE según las siglas inglesas):

$$NMSE = \sqrt{\frac{\sum_{\eta=1}^q \sum_{\xi=1}^N [y_{\xi\eta} - F_{\eta}(\mathbf{x}_{\xi})]^2}{\sum_{\eta=1}^q \sum_{\xi=1}^N [y_{\xi\eta} - \bar{y}_{\eta}]^2}} \quad (3.62)$$

donde $F_{\eta}(\mathbf{x}_{\xi})$ es la salida η -ésima de la RBFNN del vector de entrada \mathbf{x}_{ξ} , e \bar{y}_{η} es la media de los datos de salida deseados de la coordenada η -ésima. Este tipo de error permite una estimación fiable de los errores relativos gracias a la inclusión de las diferencias con respecto a la media de los datos de salida. A pesar de la extracción de la singularidad, las funciones de Green presentan cierta variación en sus valores dependiendo de la distancia entre punto observación y punto fuente. Si se utiliza un tipo de error absoluto para medir el rendimiento de la RBFNN se puede llegar a conclusiones equivocadas. Así, en una zona del espacio transversal el error absoluto puede ser pequeño debido únicamente a que los valores de las funciones de Green también lo son. Sin embargo, en puntos cercanos a la fuente el error absoluto puede ser elevado sin que ello signifique necesariamente que la aproximación posea baja calidad. Este hecho puede ser debido a que las funciones de Green toman valores mayores cerca de la fuente en comparación con otras zonas del espacio. Por tanto, la utilización de una medida de los errores relativos constituye en este caso una medida más apropiada que una medida absoluta del error en la tarea de aproximación de las funciones de Green.

Una vez determinados los parámetros que controlan el funcionamiento de los tres algoritmos de entrenamiento y el tipo de error, las estrategias supervisadas y no supervisadas pueden ser empleadas en la aproximación de las funciones de Green. Para todas las pruebas realizadas la frecuencia se fijó en un valor de 2.5 GHz. En el primer test realizado se ha utilizado la cavidad sin tapas superior ni inferior. El error NMSE de entrenamiento y generalización se evalúa cada 10 neuronas hasta que la RBFNN alcanza un tamaño total de 700 neuronas. En las Figuras 3.18 y 3.19 se observa el error NMSE para los conjuntos de entrenamiento y validación. Los resultados del algoritmo de selección aleatoria y de agrupación de centros en K-medias (“K-means”) se obtienen de la media de 10 ejecuciones. El algoritmo OLS ha sido utilizado con diversos valores de varianza. Como se aprecia en ambas figuras, el rendimiento de este algoritmo supervisado depende en gran medida del valor de varianza fija-

do. En este caso, los valores de varianza mayores arrojan mejores resultados. Hay que destacar que en ninguno de los algoritmos la capacidad de generalización se degrada por sobre-entrenamiento. Este efecto es producido por el aumento excesivo del número de neuronas respecto al tamaño del conjunto de entrenamiento [Niyogi y Girosi, 1996].

La ausencia de fuertes variaciones en las funciones de Green cuando no existen tapas superiores ni inferiores y tras la extracción de la singularidad posibilita una generalización de calidad similar a la aproximación realizada. Debido a las favorables características de las funciones de Green los algoritmos no supervisados alcanzan un nivel bajo de error. No obstante, la tasa de descenso en el error de dichos algoritmos es menor que en el algoritmo OLS con varianza 0,10. Además, no alcanzan los niveles de error mínimos conseguidos por el algoritmo OLS con varianza igual a 0,10 y a 0,050. De este modo, la selección de centros aleatoria y el algoritmo K-medias requerirán un mayor número de neuronas para alcanzar un determinado nivel de error máximo. Por lo tanto, siempre que se fije un valor adecuado de varianza el algoritmo OLS ofrecerá RBFNNs más compactas, que aceleren en mayor medida el análisis de los circuitos encapsulados.

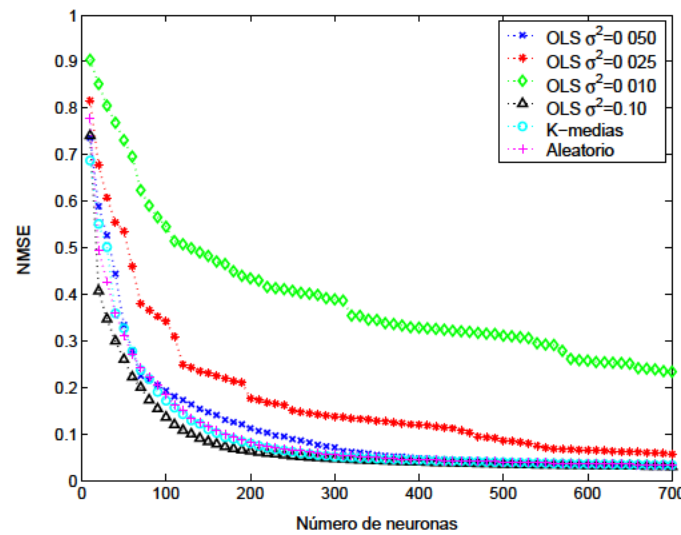


Figura 3.18: Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de entrenamiento cuando la cavidad de la Figura 3.7 no posee tapa superior ni inferior.

Para observar con más detalle el rendimiento de las RBFNNs entrenadas con distintos algoritmos se muestran en las Figuras 3.20 los valores de salida de las RBFNNs y de la parte real de G_V para dos puntos fuente. Uno de los puntos fuente se ubica lejos de las paredes mientras que el otro se localiza en las proximidades de las paredes. Para ambos casos los puntos observación se sitúan en una recta que pasa por el punto fuente (ver Figura 3.7). En 3.20(a) se observa que el rendimiento del mejor algoritmo OLS y de la estrategia de agrupación de centros es similar cuando el punto fuente está en el centro de la cavidad. Cuando la varianza toma un valor demasiado pequeño el algoritmo OLS no es capaz de seguir adecuadamente las variaciones de la parte real de G_V . En cambio, cerca de las paredes las funciones de Green presentan un mayor crecimiento, de modo que cerca de la fuente una varianza menor parece más adecuada. Esto es debido a que una menor varianza produce gaussianas más estrechas, con un crecimiento rápido de la función gaussiana cerca de su centro. Por ello, este

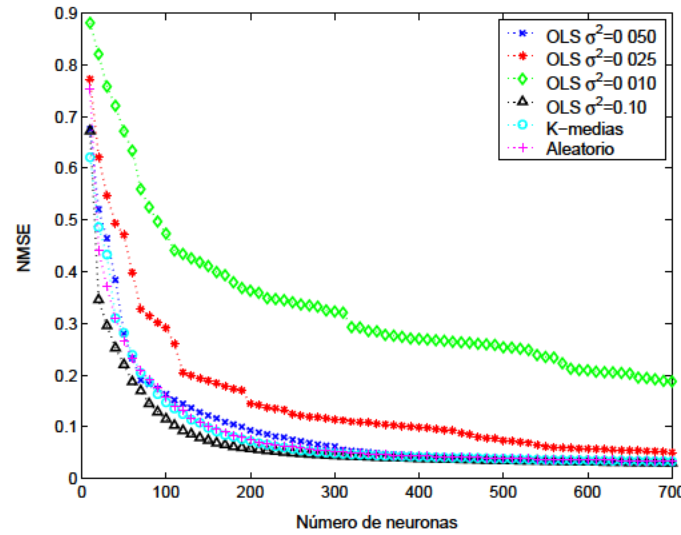
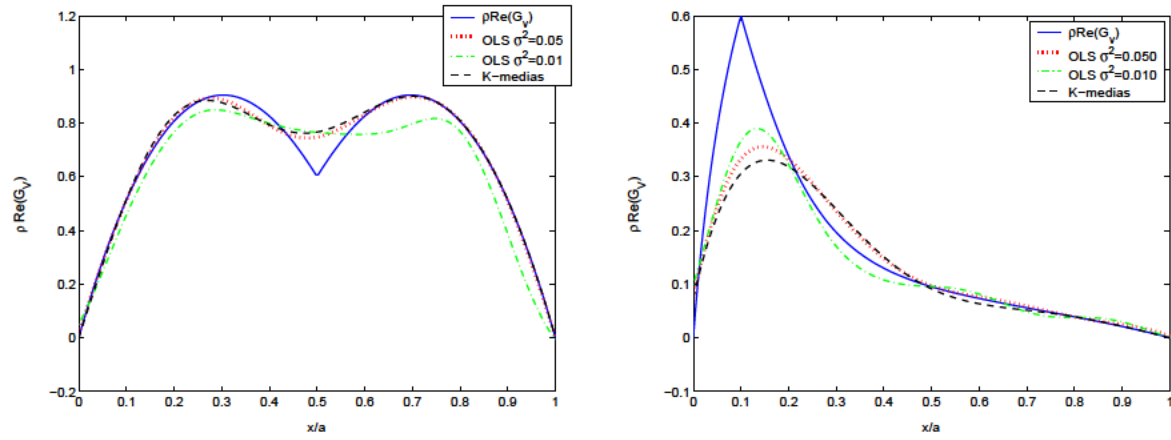


Figura 3.19: Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de validación cuando la cavidad de la Figura 3.7 no posee tapa superior ni inferior.

tipo de gaussianas son más adecuadas para aproximar funciones de rápido crecimiento. A pesar de la ligera mejora cerca de las paredes lograda con valores de varianza pequeños, la degradación de la aproximación lejos de los muros metálicos desaconseja su utilización frente a otros valores de varianza mayores. El algoritmo OLS con varianza 0,05 consigue un resultado ligeramente mejor que el algoritmo de K-medias. La pequeña diferencia en el nivel de error NMSE entre los mejores algoritmos OLS y el de agrupación de centros se debe a la mejor capacidad de aproximación por parte del algoritmo OLS en este tipo de situaciones. En [Garcia et al., 2004c] y [Garcia et al., 2004b] se muestran resultados similares para G_V sobre una estructura diferente. En estos trabajos se llega asimismo a la conclusión de que el algoritmo OLS alcanza un mejor rendimiento que los algoritmos no supervisados.

En el segundo test realizado, los puntos fuente y observación se encuentran en el interior de la cavidad totalmente apantallada. En este caso, el error NMSE de aproximación y generalización se muestra cada 20 neuronas, hasta llegar a un total de 1000 neuronas. En la Figura 3.21 se muestran los niveles de error NMSE del conjunto de entrenamiento mientras que en la Figura 3.22 se dibujan los niveles del conjunto de validación. Al igual que en el primer test, el algoritmo OLS ha sido ejecutado varias veces con distintos valores de varianza. Los niveles de error de los algoritmos no supervisados corresponden a la media de 10 ejecuciones. La primera conclusión, a la vista de los niveles de error, es que la aproximación de las funciones de Green se torna mucho más difícil en la cavidad totalmente apantallada. Este hecho, es debido a que el comportamiento de las funciones de Green es ahora más abrupto tanto cerca como lejos de las paredes, tal como se observa en las Figuras 3.14 y 3.15, en comparación con el caso de la cavidad sin tapas. A pesar de la extracción de la singularidad el crecimiento de las funciones de Green cerca de la fuente impide una aproximación adecuada. El algoritmo OLS con una varianza igual a 0,10 consigue el menor error NMSE de entrenamiento pero en cambio posee el segundo peor registro de error de generalización para 1000 neuronas. En este caso particular del algoritmo OLS y en el algoritmo de selección aleatoria aparece el efecto negativo del



(a) Punto fuente situado en el centro de la caja. Los puntos observación se encuentran en la línea $y/b=0.50$ que pasa sobre el punto fuente.

(b) Punto fuente situado cerca de las paredes de la caja. Los puntos observación se encuentran en la línea $y/b=0.10$ que pasa sobre el punto fuente.

Figura 3.20: Aproximación de la parte real de G_V una vez eliminada la singularidad para dos puntos fuente de la estructura 3.7 sin tapas superior ni inferior.

sobre-entrenamiento. A partir de un cierto número de neuronas, la RBFNN pierde la capacidad de generalización y consecuentemente es incapaz de aproximar correctamente los datos de validación. Los mejores resultados tanto en el conjunto de entrenamiento como en el de validación corresponden al algoritmo OLS con un valor de varianza de 0,025. Es un valor suficientemente pequeño para aproximar en cierta medida el rápido crecimiento cerca de la fuente pero sin impedir la aproximación de las variaciones de las funciones de Green lejos de la fuente. El algoritmo no supervisado de agrupación de centros evita el efecto del sobre-entrenamiento pero no consigue alcanzar los niveles de error del algoritmo OLS. Los datos de entrenamiento se distribuyen en una rejilla regular de puntos, sin grupos diferenciados de puntos, de modo que el algoritmo de K-medias no presenta una ventaja sustancial frente a un algoritmo supervisado.

Después del análisis de los resultados se debe elegir el algoritmo OLS como método de entrenamiento de las RBFNNs necesarias en la tarea de aproximación de las funciones de Green. Tanto en las Figuras de error 3.21 y 3.22 como en 3.18 y 3.19 se observa que en el valor de NMSE de validación es en ocasiones menor que el error NMSE de entrenamiento. Habitualmente, el error de generalización es mayor que el de entrenamiento. En ocasiones puede darse este fenómeno especialmente si existen pocas neuronas en la red.

Para ilustrar el rendimiento de los algoritmos de entrenamiento se ha simulado la respuesta de la RBFNN para tres casos concretos. Así, en la Figura 3.23 se muestran los valores de salida neuronal y la parte real de G_V para dos posiciones diferentes de un punto fuente. Los puntos observación asociados se sitúan en cortes que pasan por el punto fuente como se observa en la Figura 3.7. Las respuestas ofrecidas por los tres algoritmos no son capaces de seguir el rápido crecimiento de la función de Green cerca de la fuente. La mejor aproximación en esta zona corresponde para el algoritmo OLS con el menor valor de varianza tanto lejos de las paredes (Figura 3.23(a)) como cerca de las paredes (Figura 3.23(b)). Como ya se explicó en el caso de la cavidad sin tapas, un menor valor de varianza produce gaussianas de variación más rápida cerca del centro y por tanto constituyen fun-

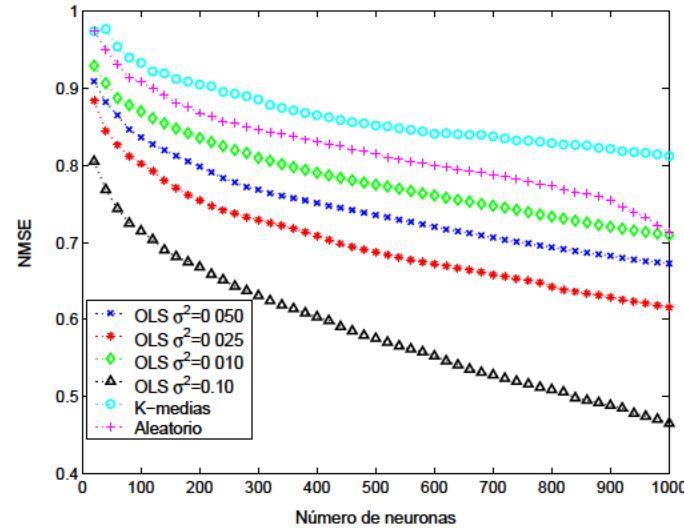


Figura 3.21: Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de entrenamiento en la cavidad de la Figura 3.7 apantallada totalmente.

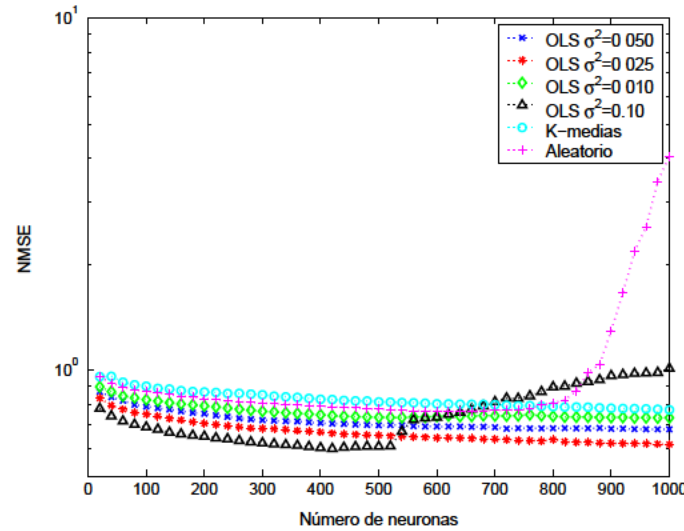


Figura 3.22: Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de validación en la cavidad de la Figura 3.7 apantallada totalmente.

ciones neuronales más aptas para aproximar los cambios abruptos cerca de la fuente. Sin embargo, lejos de la fuente la función de Green presenta un comportamiento menos abrupto y por tanto un valor pequeño de varianza impide una aproximación correcta. El valor de varianza de 0,025 representa así una solución de compromiso entre las variaciones de las funciones de Green cerca y lejos de la fuente. Por ello, consigue un nivel de error NMSE general menor que el alcanzado cuando la varianza es fijada a 0,010.

Debido a la influencia del apantallamiento total, las funciones de Green exhiben un comportamiento similar en el crecimiento cerca de la fuente. La extracción de la singularidad empleada suaviza las variaciones en todo el plano transversal pero el crecimiento en el entorno de la fuente es

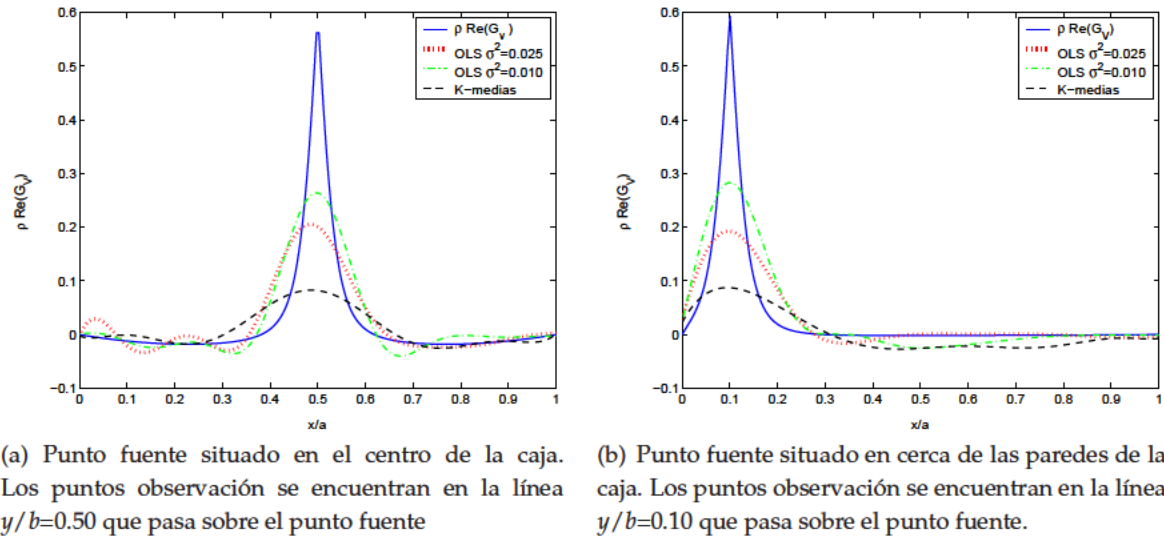


Figura 3.23: Aproximación de la parte real de G_V una vez eliminada la singularidad para dos puntos fuente de la estructura 3.7 totalmente apantallada.

todavía significativo. Por lo tanto, la rejilla utilizada para generar el conjunto de entrenamiento no permite que las funciones de Green sean correctamente aproximadas por las RBFNNs aun empleando un algoritmo supervisado.

Finalmente, terminamos esta Sección con la exposición de los resultados referidos al tiempo de cálculo para el caso de apantallamiento total. En la Tabla 3.1 se muestran el tiempo de entrenamiento de los tres algoritmos para diferentes cantidades de neuronas. Asimismo, se incluye el tiempo empleado por la RBFNN para generar los valores de salida para los datos de entrenamiento y validación y el tiempo invertido en generar los conjuntos anteriores mediante las técnicas explicadas en el Capítulo 2. La ganancia temporal conseguida por la RBFNN se cifra en hasta dos órdenes de magnitud cuando la red neuronal está compuesta por un número reducido de neuronas. Incluso si la RBFNN cuenta con un número elevado de neuronas la ganancia temporal es relevante. El algoritmo OLS requiere una cantidad de tiempo muy superior a los otros dos algoritmos para generar un número determinado de neuronas. Además, necesita almacenar en memoria la matriz \mathbf{P} descrita en la ecuación (3.21). De esta manera, el número de datos de entrenamiento tiene un límite máximo por encima del cual el entrenamiento con la estrategia OLS es imposible debido a las limitaciones de memoria impuestas por el ordenador. El tiempo invertido por el algoritmo OLS puede llegar a producir que el método neuronal sea impracticable. De todos modos, se ha optado por este algoritmo ya que permite una aproximación mejor de las funciones de Green, tal como muestran las Figuras 3.21 y 3.22 y como mostrarán los resultados de secciones siguientes. Este método supervisado consigue redes neuronales más compactas y en consecuencia una mayor ganancia temporal que convierte en altamente rentable el tiempo invertido en el entrenamiento. Además, en la Sección 3.9 se desarrolla una mejora del algoritmo OLS que reduce drásticamente el tiempo empleado en el entrenamiento neuronal.

Tabla 3.1: Resultados en tiempo de cálculo para los algoritmos de entrenamiento estudiados. Todas las simulaciones se llevaron a cabo con un ordenador con un procesador Pentium IV a 3GHz y con 4GB de memoria RAM.

Número de neuronas	Algoritmo de entrenamiento (s)			Salida RBFNN (s)	Salida exacta (s)
	OLS	K-medias	Aleatorio		
701	2509	32.1	29.7	2.42	85.3
501	1784	17.5	15.6	1.75	85.3
301	107	7.7	6.5	1.15	85.3
101	365	1.8	1.3	0.53	85.3

3.6. Comparativa entre RBFNN, MLP y SVM en la aproximación de las funciones de Green

En la Sección anterior se ha comparado el rendimiento de tres algoritmos diferentes de entrenamiento de una RBFNN en un problema de aproximación de funciones de Green. Las pruebas realizadas confirmaron que el algoritmo supervisado, gracias al uso de los valores de salida en el cálculo de los parámetros neuronales, es capaz de alcanzar mayores niveles de precisión que los algoritmos no supervisados. En la Sección 3.2 se han explicado las razones que han llevado a la selección de la Redes neuronales de funciones base radial para sustituir a las funciones de Green en la evaluación de la IE. Ahora bien, las redes neuronales como las de clase MLP o las SVM, descartadas para aproximar las funciones de Green, utilizan métodos de entrenamiento supervisados. Tanto los MLP como las SVM han demostrado su eficiencia en diversos problemas de regresión y clasificación. Por lo tanto, es adecuado repetir un test como el realizado en la comparativa entre métodos de entrenamiento de RBFNN, para confirmar que las MLP y las SVM no cumplen los requisitos requeridos para la aceleración de análisis circuital. Al igual que la Sección 3.5 se ha empleado la estructura de puntos de entrenamiento y validación dibujada en la Figura 3.16. Asimismo, se utiliza la misma cavidad de la Figura 3.7 de modo que los conjuntos de entrenamiento y validación son iguales a los usados en la anterior Sección. Ambas redes neuronales, MLP y SVM, cuentan con cuatro entradas correspondientes a las coordenadas espaciales de los puntos fuente y observación. El espacio de salida está formado por las partes reales e imaginarias de las funciones de Green con cancelación de la singularidad realizada según el método descrito en la Sección 3.4. La frecuencia se fijó en todas las pruebas a 4 GHz.

El primer test emplea la red neuronal MLP para aproximar las funciones de Green de la cavidad 3.7 en dos casos diferentes. En un primer caso, en la cavidad sin tapas metálicas superior ni inferior y en un segundo caso en el que el apantallamiento es total. En cada caso el MLP es diseñado con siete algoritmos de entrenamiento diferentes disponibles en la toolbox de redes neuronales de *MATLAB*® (versión 7.14). Este programa presenta una amplia gama de algoritmos de entrenamiento para este tipo de redes neuronales. En particular, se han empleado los siguientes algoritmos: Levenberg-Marquadt, Quasi-Newton con el Hessiano calculado mediante la fórmula Broyden-Fletcher-Goldfarb-Shanno (BFGS), gradiente conjugado escalado, gradiente conjugado Powell/Beale, gradiente conjugado Fletcher-Powell, gradiente conjugado Polak-Ribiere y Quasi-Newton con el método de la secante. Los algoritmos de gradiente conjugado emplean direcciones

conjugadas del gradiente, en lugar de una única dirección negativa, para converger a la solución. Las diferencias entre cada uno de los métodos se basan en las distintas formas de actualizar la cuantía del incremento de los pesos del MLP. El algoritmo de Quasi-Newton emplea el Hessiano para calcular la variación que deben sufrir los pesos en cada iteración. En el primer algoritmo de este tipo se utiliza la formula BFGS para actualizar el Hessiano. En el segundo caso, el Hessiano se actualiza mediante el método de la secante. Este mecanismo permite una reducción del tiempo de calculo y de la memoria consumida respecto a la fórmula BFGS. De este modo, representa un algoritmo intermedio respecto a los métodos de gradiente conjugado y el algoritmo Quasi-Newton con BFGS. Finalmente, el algoritmo de entrenamiento Levenberg-Marquadt se fundamenta en la aproximación del Hessiano mediante el calculo del Jacobiano. Los resultados conseguidos con otros algoritmos programados en la toolbox de *MATLAB*® basados en mecanismos de retro-propagación no se muestran debido al elevado nivel de error alcanzado en la aproximación. Para todos los casos, el MLP está constituido por una única capa intermedia de neuronas. La función no lineal utilizada en cada neurona es de tipo tangente hiperbólica.

En las Figuras 3.24 se exponen los niveles de error NMSE cuando la estructura multicapa de la Figura 3.7 no posee tapas superior ni inferior. Los resultados para cada algoritmo son la media de cinco ejecuciones con pesos iniciales diferentes en cada una de las ejecuciones. Excepto en el algoritmo Levenberg-Marquadt, todos los demás algoritmos son muy imprecisos en la aproximación de las funciones de Green. Los niveles de error son muy superiores a los logrados en general con las RBFNNs, ver Figuras 3.18 y 3.19. El algoritmo Levenberg-Marquadt consigue una precision ligeramente inferior a la alcanzada con las RBFNNs. No obstante, la necesidad de almacenamiento del Jacobiano y de otras matrices impide el aumento del número de neuronas. La cantidad de memoria para guardar las mencionadas matrices resultó insuficiente. Hay que resaltar que las pruebas se realizaron en un ordenador con una memoria RAM de 4 GB. Además, el tiempo de cómputo de los parámetros es muy elevado en este algoritmo, superando incluso ampliamente al algoritmo OLS en este aspecto. Por ejemplo, para entrenar un MLP de 100 neuronas el algoritmo de Levenberg-Marquadt necesita casi seis horas. El tiempo de entrenamiento aumenta hasta las 13 horas si el MLP está formado por 200 neuronas.

Si la cavidad se encuentra apantallada totalmente la aproximación de las funciones de Green es más difícil debido a las rápidas variaciones cerca de la fuente. Como se vio en las Figuras 3.18 y 3.19 el error NMSE de las RBFNNs aumenta ampliamente cuando se colocan tapas superior e inferior en la estructura. De un modo similar, la aproximación que realiza la red neuronal MLP se convierte en muy imprecisa en la situación de apantallamiento, tal como se observa en las Figuras 3.25(a) y 3.25(b). El algoritmo Levenberg-Marquadt alcanza los menores niveles de error de entrenamiento. Sin embargo, este algoritmo presenta elevados niveles de error de validación al igual que el resto de métodos de entrenamiento. Los resultados muestran que la red neuronal MLP no es capaz de alcanzar la precisión lograda por la RBFNN en ninguna de las dos situaciones de apantallamiento estudiadas. Una segunda desventaja de la red neuronal MLP frente a la RBFNN radica en el elevado tiempo de entrenamiento requerido por el MLP.

Para confirmar los resultados sobre el MLP obtenidos con las herramientas de *MATLAB*® se repitieron las pruebas realizadas con otra herramienta de entrenamiento de este tipo de red neuronal. *Netlab*® (disponible en <http://www.ncrg.aston.ac.uk/netlab/index.php>) es un conjunto de rutinas

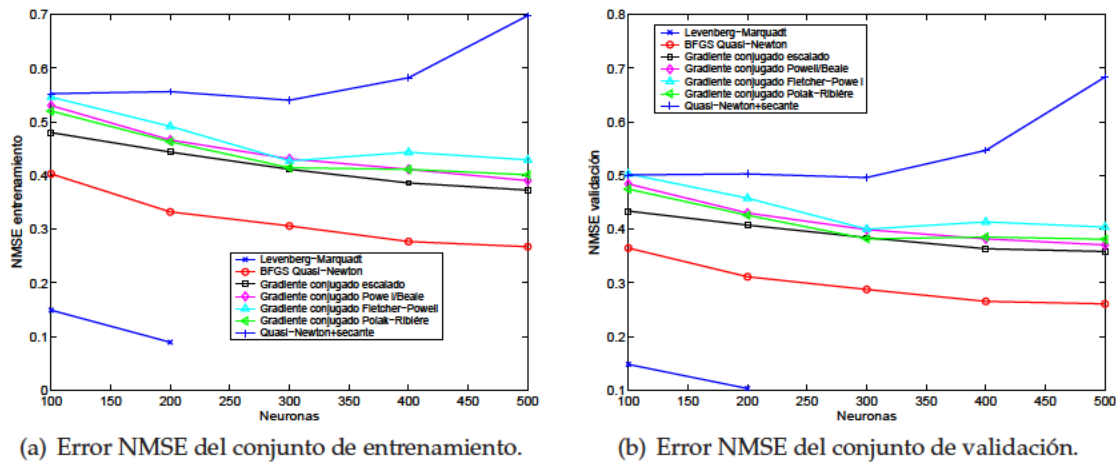


Figura 3.24: Error NMSE para un MLP diseñado con siete algoritmos de entrenamiento diferentes. La estructura de trabajo corresponde a la cavidad dibujada en la Figura 3.7 sin tapas superior ni inferior.

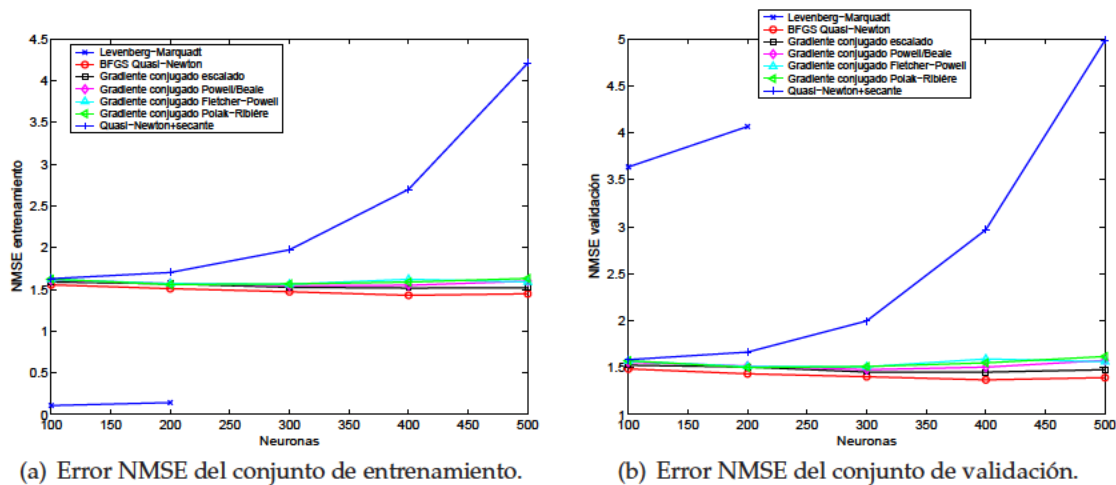


Figura 3.25: Error NMSE para un MLP diseñado con siete algoritmos de entrenamiento diferentes. La estructura de trabajo corresponde a la cavidad dibujada en la Figura 3.7 totalmente apantallada.

de entrenamiento escritas en lenguaje *MATLAB*® y que han sido utilizadas ampliamente en diversos trabajos científicos. Se han empleado tres métodos de entrenamiento: gradiente conjugado, gradiente conjugado escalado y Quasi-Newton. En las Tablas 3.2 y 3.3 se muestran los niveles de error de entrenamiento y validación para las dos situaciones de apantallamiento estudiadas. En todos los algoritmos de entrenamiento se aplicaron 1500 iteraciones. En el caso de la caja sin tapas superior ni inferior la precisión es baja excepto en el algoritmo que utiliza iteraciones de tipo Quasi-Newton. Sin embargo, el tiempo de entrenamiento en este método es muy grande (más de ocho horas). En el caso de total apantallamiento los algoritmos de gradiente conjugado presentan niveles de error muy elevados. El algoritmo basado en Quasi-Newton consigue un rendimiento algo mejor en el conjunto de entrenamiento pero la precisión es baja en el conjunto de validación. Además, de nuevo el algoritmo es muy lento ya que requiere más de siete horas para realizar las iteraciones fijadas. Todas las simu-

laciones con la red neuronal MLP fueron realizadas con un ordenador con un procesador Pentium IV a 3GHz, 4 GB de RAM y la versión 7.14 de *MATLAB*®.

Tabla 3.2: Niveles NMSE de entrenamiento y validación para la cavidad sin tapas superior ni inferior.

	Número de neuronas									
	100		200		300		400		500	
Algoritmo	Ent.	Val.	Ent.	Val.	Ent.	Val.	Ent.	Val.	Ent.	Val.
Grad. conj.	0.746	0.664	0.658	0.588	0.640	0.575	0.641	0.565	0.709	0.628
Grad. conj. esc.	0.625	0.563	0.632	0.565	0.606	0.537	0.626	0.560	0.643	0.567
Quasi-Newton	0.126	0.125								

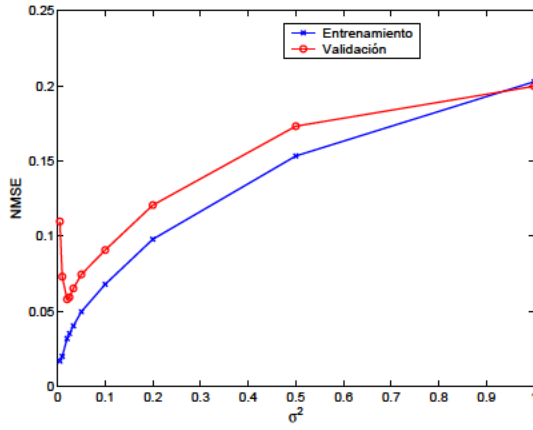
Tabla 3.3: Niveles NMSE de entrenamiento y validación para la cavidad totalmente apantallada.

	Número de neuronas									
	100		200		300		400		500	
Algoritmo	Ent.	Val.	Ent.	Val.	Ent.	Val.	Ent.	Val.	Ent.	Val.
Grad. conj.	1.655	1.614	1.665	1.630	1.660	1.622	1.664	1.628	1.660	1.620
Grad. conj. esc.	1.655	1.614	1.663	1.626	1.658	1.620	1.663	1.626	1.653	1.614
Quasi-Newton	0.957	1.730								

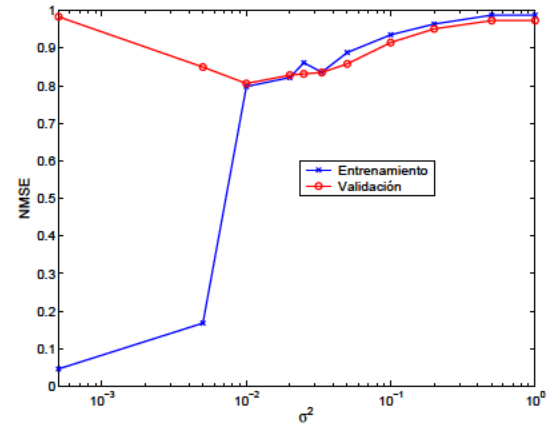
En el segundo test la red neuronal SVM es empleada en la aproximación de las funciones de Green tanto en la cavidad apantallada como en la cavidad sin tapas superior ni inferior. El algoritmo elegido para entrenar la red neuronal es el conocido como SVM-Light, escrito en lenguaje C y disponible en internet (<http://svmlight.joachims.org>). El mencionado algoritmo es uno de los mas eficientes y veloces en seleccionar los vectores soporte entre los datos de entrenamiento [Joachims, 1999, Joachims, 2002]. En las Figuras 3.26(a) y 3.26(b) se exhibe el error NMSE de entrenamiento y validación para la situación de apantallamiento parcial y total, respectivamente. El error se evaluó para diferentes valores de varianza. El valor de varianza es común e igual para todas la neuronas. Asimismo, se realizaron diversas pruebas con distintos valores de los parámetros ϵ y C de la SVM. El valor de ϵ mide el máximo error permitido en la aproximación mientras que C representa la relación de compromiso entre el error de entrenamiento y la complejidad de la red neuronal. Los resultados mostrados en la Figura 3.26 corresponden a los mejores obtenidos en todas las pruebas realizadas con los diferentes valores asignados a ϵ y C .

Al igual que en el caso de la RBFNN, existe una serie de valores de varianza que maximizan la capacidad de generalización. A partir de cierto límite tanto la aproximación en el conjunto de entrenamiento como la capacidad de generalización se degrada. El error NMSE es mucho mayor en la situación de total apantallamiento que en el caso de apantallamiento parcial. Cuando la varianza se selecciona de forma apropiada, el nivel de error de entrenamiento y validación es similar al conseguido con las RBFNNs entrenadas con el algoritmo OLS.

En el problema de la aproximación de las funciones de Green es fundamental que las redes neuronales posean un número reducido de neuronas. De esta manera, cuando las redes neuronales susti-



(a) Error NMSE del conjunto de entrenamiento y validación cuando la caja no posee tapas superior ni inferior.



(b) Error NMSE del conjunto de entrenamiento y validación cuando la caja está totalmente apantallada.

Figura 3.26: Error NMSE en una SVM para diferentes valores de varianza.

tuyan a las funciones de Green en la resolución de la IE, el tiempo total de análisis será disminuido. Sin embargo, para alcanzar cierta precisión las SVM requieren de un número muy elevado de vectores soporte, es decir, de neuronas. En [Osowski y Markiewicz, 2005] se realiza una comparación entre el rendimiento del algoritmo de entrenamiento OLS y del método SVM para entrenar RBFNNs. El trabajo anterior muestra que si no aparece el problema del sobre-entrenamiento el algoritmo OLS requiere, para los ejemplos utilizados en el artículo, un número de neuronas menor que el algoritmo SVM para alcanzar niveles de error semejantes. En las Figuras 3.27 se exponen el número de vectores soporte para las dos situaciones de apantallamiento en la estructura multicapa de trabajo. En ambos casos la cantidad de neuronas es de varios miles, es decir, hasta un orden de magnitud mayor que el número de neuronas requerido en las RBFNNs. Por lo tanto, el análisis será acelerado en una medida mucho mayor si se emplean RBFNNs que en el caso de utilizar SVMs. Esta es una razón suficiente para descartar a las SVM como herramienta para aproximar las funciones de Green del medio apantallado multicapa.

3.7. Algoritmo de división del espacio de entrada. Varianza fija

En la Sección 3.5 se ha podido comprobar la dificultad de la aproximación de las funciones de Green del medio apantallado multicapa. Como se aprecia en las Figuras 3.21 y 3.22 una red regular de puntos fuente y observación no es suficiente para expresar con corrección las variaciones de las funciones de Green de la cavidad bajo estudio. Incluso en el caso de que la red de puntos cubra totalmente el plano transversal de la cavidad, el error de aproximación y generalización no alcanza los niveles mínimos para desarrollar una herramienta de análisis neuronal. Una solución posible consiste en aumentar el número de puntos fuente y observación de la red regular. De esta forma, las funciones de Green serían representadas de manera más completa, tanto en las cercanías de la fuente como cerca de las paredes metálicas. No obstante, si se emplea una red de puntos de entrenamiento muy tupida, el entrenamiento se puede volver imposible. En el ejemplo práctico de la Figura 3.16 el conjunto de entrenamiento estaba formado por 8281 muestras. Cada una de estas muestras estaba

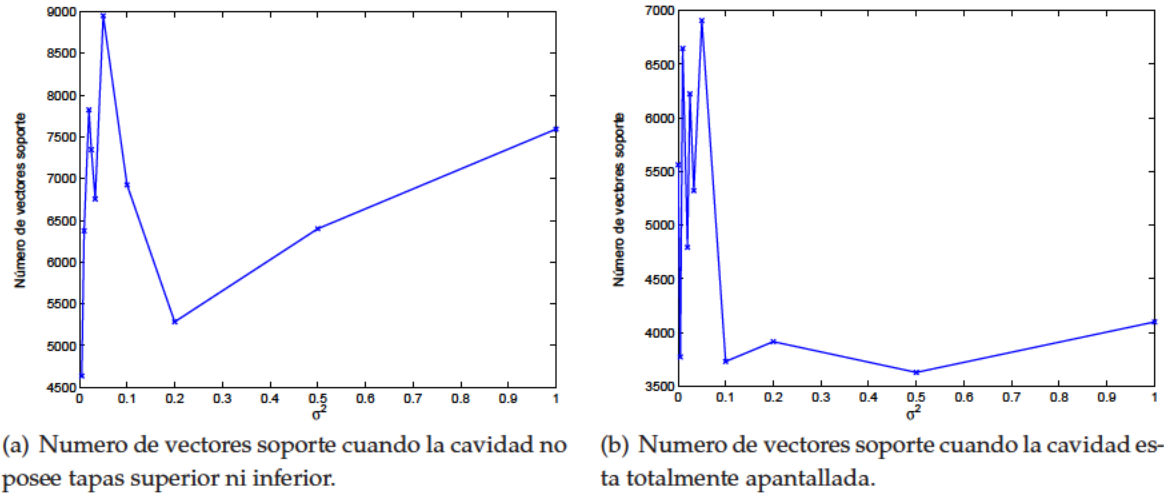


Figura 3.27: Número de vectores soporte para diferentes valores de varianza.

formada por cuatro coordenadas espaciales de forma que no se incluía la variación en la frecuencia. Sin embargo, si se desea que las RBFNNs actúen en una formulación real de la IE es necesario que las funciones de Green sean aproximadas en un determinado ancho de banda. Si se añade una coordenada frecuencial y es muestreada en tres puntos, el conjunto total de entrenamiento sería igual a 24843. El algoritmo OLS necesita almacenar la matriz \mathbf{P} de regresores de tamaño $N \times N$, con $N = 24843$ igual al tamaño del conjunto de entrenamiento. Por tanto, el entrenamiento con el algoritmo supervisado OLS es imposible si, por ejemplo, el ordenador en el que se realiza la simulación tiene una memoria RAM inferior a 4,60 GB, y los datos se guardan en un formato de doble precisión de 64 bits como el existente en el lenguaje de programación Fortran [Chapman, 1998]. El incremento en el número de puntos del conjunto de entrenamiento conduce a la superación de los límites de la memoria del ordenador.

Los algoritmos no supervisados como el de selección aleatoria o el de agrupación en K-Medias no necesitan el almacenamiento de matrices o vectores que saturan la memoria del computador. Por tanto, pueden emplearse con conjuntos de entrenamiento de gran tamaño. No obstante, si el conjunto de entrenamiento crece, el número de neuronas que la RBFNN necesita para aproximar los datos crecerá asimismo. Si el tamaño de la RBFNN crece por encima de un límite, el método de análisis neuronal pierde su atractivo debido al aumento de tiempo de análisis. El número de neuronas no puede ser tal que la generación de los datos de salida sea más lento que la evaluación numérica de las funciones de Green. Tanto si se utilizan algoritmos de entrenamiento que requieran muchos recursos de computación como si no necesitan memoria o tiempo de cálculo, el conjunto de entrenamiento no puede sobrepasar ciertos límites. La red de puntos fuente y observación de la Figura 3.16 no puede ser empleada para la tarea de aproximación de las funciones de Green.

La solución propuesta al anterior problema radica en aplicar el principio de dividir y vencer. Para lograr una adecuada capacidad de aproximación y generalización, el espacio de entrada de la RBFNN es dividido en distintas zonas. Esta estrategia ha sido empleada con éxito en otros problemas de modelado electromagnético resueltos mediante el uso de redes neuronales. Por ejemplo, en [Devabhaktuni et al., 2001] el espacio de entrada de la red neuronal es dividido en zonas y muestrea-

do con mayor intensidad allí donde se hallan los mayores niveles de error. En nuestro problema será necesario dividir la región espacial de los puntos observación y fuente.

Los puntos observación son agrupados dependiendo de su proximidad al punto fuente. Aunque la singularidad es extraída, el rápido crecimiento de las funciones de Green impone que las zonas cerca de la fuente sean de menor tamaño que aquellas situadas lejos de la fuente. Si se emplea una cantidad de puntos observación similar en cada zona observación, entonces en el entorno de la fuente la densidad de puntos será mayor que en aquellas zonas alejadas del punto fuente (ver Figura 3.28). Una mayor densidad permite que las rápidas variaciones de las funciones de Green sean expresadas con la misma precisión que las variaciones lentas de las regiones del espacio alejadas de la fuente. Además de la división realizada sobre las coordenadas del punto observación (x, y) del espacio se realiza una segunda división sobre las coordenadas del punto fuente (x', y') . En efecto, los puntos fuente son divididos en diversas zonas dependiendo de su cercanía a las paredes metálicas. El motivo para esta separación radica en la necesidad de cumplir adecuadamente con las condiciones de contorno en las paredes metálicas del encapsulado. Para adaptarse a la estructura de la cavidad rectangular las zonas de puntos fuente tienen forma de "L" como se observa en la Figura 3.29. La densidad es similar en cada una de las zonas fuente, pudiendo aumentar hasta cierto punto, en caso de encontrar dificultades en la aproximación en alguna región espacial. Finalmente, es necesario añadir a la estructura neuronal mostrada en la Figura 3.17 una quinta coordenada correspondiente a la frecuencia, como se observa en la Figura 3.30. De este modo, las RBFNNs podrán capturar las variaciones de las funciones de Green respecto de la frecuencia. La coordenada frecuencial es asimismo segmentada en diversas zonas. La división en zonas de frecuencia es útil principalmente por dos motivos. Por un lado, esta última división es especialmente útil en las cercanías de las frecuencias de resonancia de la cavidad, donde las funciones de Green exhiben fuertes variaciones respecto a esta variable. Alrededor de las frecuencias de resonancia serán necesarias zonas de menor tamaño de forma que se expresen convenientemente las fluctuaciones de las funciones de Green. Por otro lado, la división frecuencial contribuye a la limitación del tamaño de los conjuntos de entrenamiento. Así, esta división habilita un entrenamiento más veloz, capaz de generar RBFNNs más compactas.

Los conjuntos de entrenamiento se forman a partir de la combinación de las zonas en las que se divide el espacio de entrada. Así, el primer conjunto de entrenamiento estará formado por datos en cuyas coordenadas (x, y) se encuentren puntos de la primera zona observación, con puntos de la primera zona fuente en las coordenadas (x', y') , y finalmente f puntos frecuenciales de la primera zona frecuencia. Para el segundo conjunto de entrenamiento, los puntos (x, y) pueden seguir perteneciendo a la primera zona observación, mientras que los puntos fuente se situarán en la segunda zona fuente y los puntos f pertenecerán a la primera zona frecuencial. Siguiendo el proceso anterior, si existieran p zonas fuente asociadas a la primera zona observación entonces el conjunto de entrenamiento $p + 1$ estaría formado por puntos observación de la primera zona observación, puntos fuente de la primera zona y puntos de la coordenada f de la segunda zona frecuencial. Una vez cubiertas todas las combinaciones de zonas fuente y frecuencia para la primera zona observación puede comenzar la generación de los conjuntos de otras zonas de tipo observación. Así, si existieran q zonas de tipo frecuencial, el conjunto de entrenamiento $p \times q + 1$ incluiría puntos observación pertenecientes a la segunda zona observación. El proceso continuaría hasta que se cubrieran todas las posibles combinaciones de zonas observación, fuente y frecuencia. Para garantizar que el conjun-

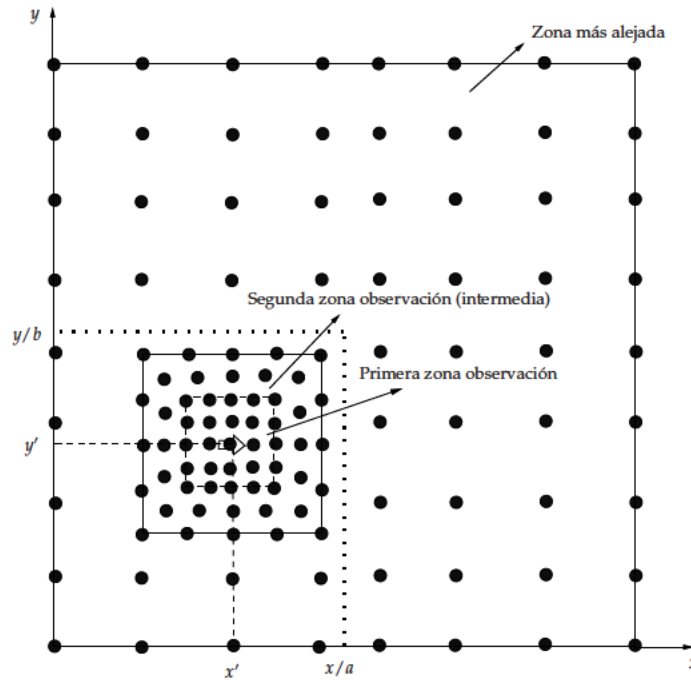


Figura 3.28: Ejemplo de zonas observación cuadradas. Los puntos observación se representan mediante puntos oscuros.

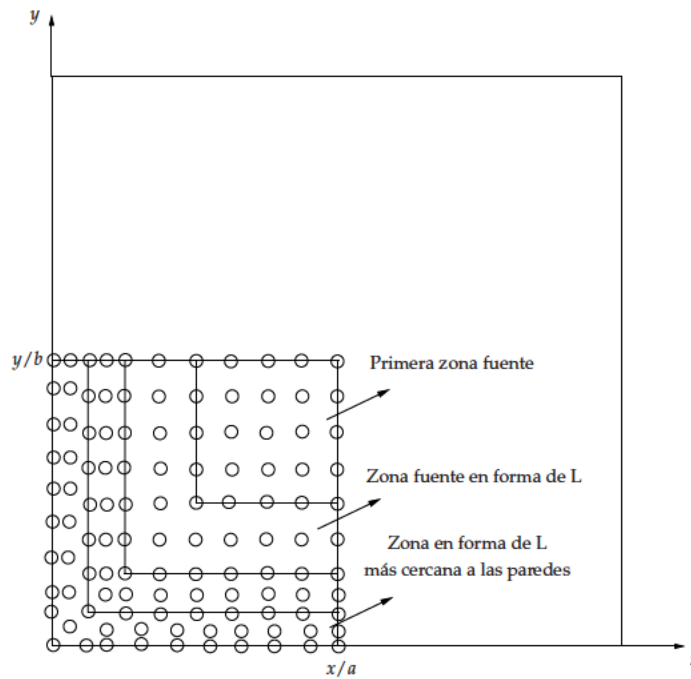


Figura 3.29: Ejemplo de zonas fuente. Los puntos fuente se representan mediante puntos transparentes.

to de entrenamiento no supera un cierto límite es necesario acotar el número de puntos que forman cada zona. Cualquier conjunto de entrenamiento tendrá así un tamaño máximo igual al producto del número máximo de elementos permitido en las zonas observación, fuente y frecuencia. De esta

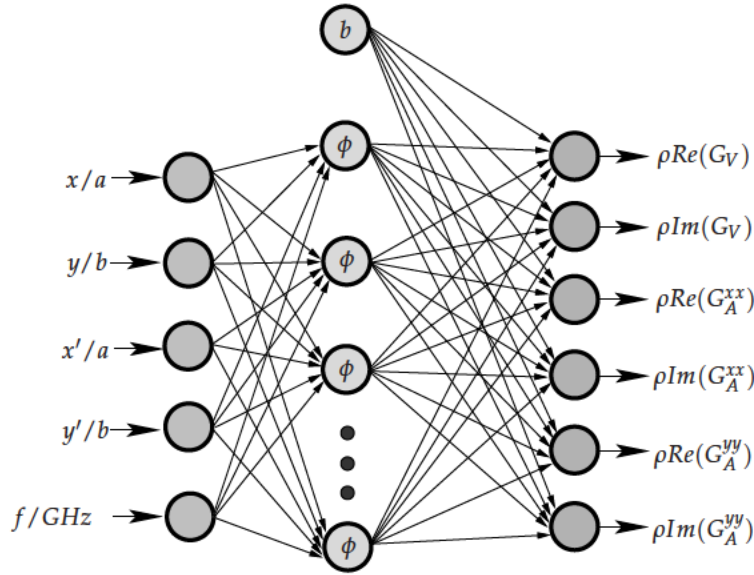


Figura 3.30: Estructura de la RBFNN que aproxima las partes reales e imaginarias de las funciones de Green multiplicadas por (ρ) en un ancho de banda de trabajo. La coordenada frecuencial del dato de entrada se expresa en GHz. Cada función gaussiana ϕ posee un centro y una varianza.

forma, se garantizará que el entrenamiento con el algoritmo OLS es posible y que la RBFNN que se encarga de aproximar las funciones de Green en esa parte del espacio de entrada no posee un número excesivo de neuronas. Si la densidad de puntos es insuficiente para alcanzar la precisión deseada se procede en primer lugar al aumento del número de puntos de las zonas espaciales y frecuenciales. En el momento en el que se supere el número máximo de elementos, la densidad se incrementará mediante la reducción del tamaño de las zonas bajo estudio.

La primera versión del algoritmo de división en zonas del espacio de entrada se basa en una rejilla regular de puntos que cubre todo el plano transversal, como la mostrada en la Figura 3.16 o la reproducida en [Garcia et al., 2004c]. Existen tres tipos de zona observación como se aprecia en la Figura 3.28. La primera zona cuadrada está compuesta por una red de puntos observación equidistribuidos alrededor de la fuente. La zonas de observación intermedias poseen una distribución de puntos observación que permite una representación adecuada de las funciones de Green. Finalmente, la zona de puntos observación más alejados de la fuente posee una distribución regular de forma que sus elementos cubren el resto del plano transversal. En esta última zona, la densidad de puntos es menor que en las zonas cercanas a la fuente debido a que en esta parte del espacio las funciones de Green presentan un comportamiento menos abrupto.

El algoritmo se inicia con la fijación de un límite que determine el final de las zonas primera e intermedias y el comienzo de la zona de puntos más alejados de la fuente. Este límite máximo puede definirse como una fracción (d_1) de la distancia entre fuentes de la red regular de puntos mostrada en la Figura 3.16. Así, la primera zona observación estaría formada por puntos distribuidos en $(x' - d_1 \leq x \leq x' + d_1, y' - d_1 \leq y \leq y' + d_1)$. Para entrenar una RBFNN es necesario además definir una serie de puntos fuente y puntos frecuencia que completen cada dato de entrada. Los puntos fuente asociados a esta primera zona observación corresponden a aquellos situados en la primera

zona fuente (ver Figura 3.29). La frecuencia se fija a un valor constante e igual al valor máximo del ancho de banda de trabajo. Las funciones de Green exhiben variaciones más rápidas cerca de la fuente en esta frecuencia. De este modo, una correcta aproximación y generalización para la zona observación en esta frecuencia asegura una aproximación eficiente en todo el ancho de banda de trabajo. Con los anteriores límites en la zona observación, zona fuente y frecuencia se genera un conjunto de entrenamiento con el que entrenar una RBFNN. El conjunto de validación asociado está compuesto por puntos situados en posiciones intermedias respecto a los puntos de entrenamiento. Si la RBFNN entrenada no supera el error máximo entonces se procede a aumentar el número de puntos observación y fuente para aumentar la densidad de puntos. En caso de no alcanzar el éxito en la aproximación se recurre a la reducción de la zona observación y si fuera necesario de la primera zona fuente. Así, la definición de la primera zona observación lleva aparejada la fijación de la primera zona fuente.

La primera zona observación presenta las mayores dificultades en la aproximación debido al rápido crecimiento cerca de la fuente. Por lo tanto, es necesaria una división en distintas zonas fuente para esta primera zona observación. De esta forma, las funciones de Green podrán ser expresadas apropiadamente cuando los puntos observación se ubiquen en las cercanías de los puntos fuente tanto lejos como cerca de las paredes. En esta primera versión del algoritmo de división en zonas no se realizó ninguna división en los puntos fuente para las zonas observación alejadas de la fuente. Por lo tanto, la zona fuente asociada a las zonas intermedias y a la zona de puntos de observación más alejados cubre todo el primer cuadrante del plano transversal. La división en regiones fuente mostrada en la Figura 3.29 se limita a la primera zona observación.

Tras la fijación los límites de la primera zona observación y de la primera zona fuente puede empezar la búsqueda del límite superior de la primera zona intermedia. Su límite inferior es el superior de la primera zona observación. En un principio la primera zona intermedia posee como límite superior el límite máximo d_1 . Al igual que en la primera zona observación, si no se alcanza el error máximo se procede al incremento de la densidad gracias al aumento de puntos o a la reducción del tamaño de la zona observación. De forma sucesiva se siguen añadiendo zonas intermedias hasta que se alcance el límite máximo d_1 .

Una vez finalizada la división de las coordenadas espaciales comienza la división de la coordenada frecuencial. La segmentación en frecuencia hace uso de la primera zona observación y de la primera zona fuente para completar las coordenadas espaciales de los datos de entrenamiento y validación. Las zonas frecuenciales generadas con las mencionadas regiones espaciales son aplicadas al resto de zonas observación intermedias y a la zona de puntos observación alejados de la fuente. La definición de las zonas frecuenciales se realiza sobre la zona de más difícil aproximación, cerca de la fuente. De este modo, se puede emplear en el resto de zonas observación sin que aparezcan altos niveles de error de entrenamiento u otros problemas como el sobre-entrenamiento. La división de la coordenada frecuencial debe tener en cuenta la existencia de frecuencias de resonancia que pueden aparecer en entornos apantallados. En la Figura 3.31 se observa la operación completa realizada por el algoritmo para segmentar en zonas la coordenada frecuencia. En caso de detectar una frecuencia de resonancia el algoritmo construye las zonas frecuencia necesarias desde el límite inferior del ancho de banda hasta la primera frecuencia de resonancia. Si existe una segunda frecuencia de resonancia se generan nuevas zonas frecuenciales entre la primera y segunda resonancias. El mismo proceso es

aplicado hasta que se definen las zonas frecuenciales situadas entre la última frecuencia de resonancia y el límite superior del ancho de banda de trabajo. Cuando todo el espacio de entrada ha sido definido comienza la generación de los conjuntos de entrenamiento y validación correspondientes a las distintas combinaciones de zonas. Cada RBFNN es diseñada mediante la adición de neuronas hasta que se rebasa el error máximo admitido tanto para el conjunto de entrenamiento como para el de validación.

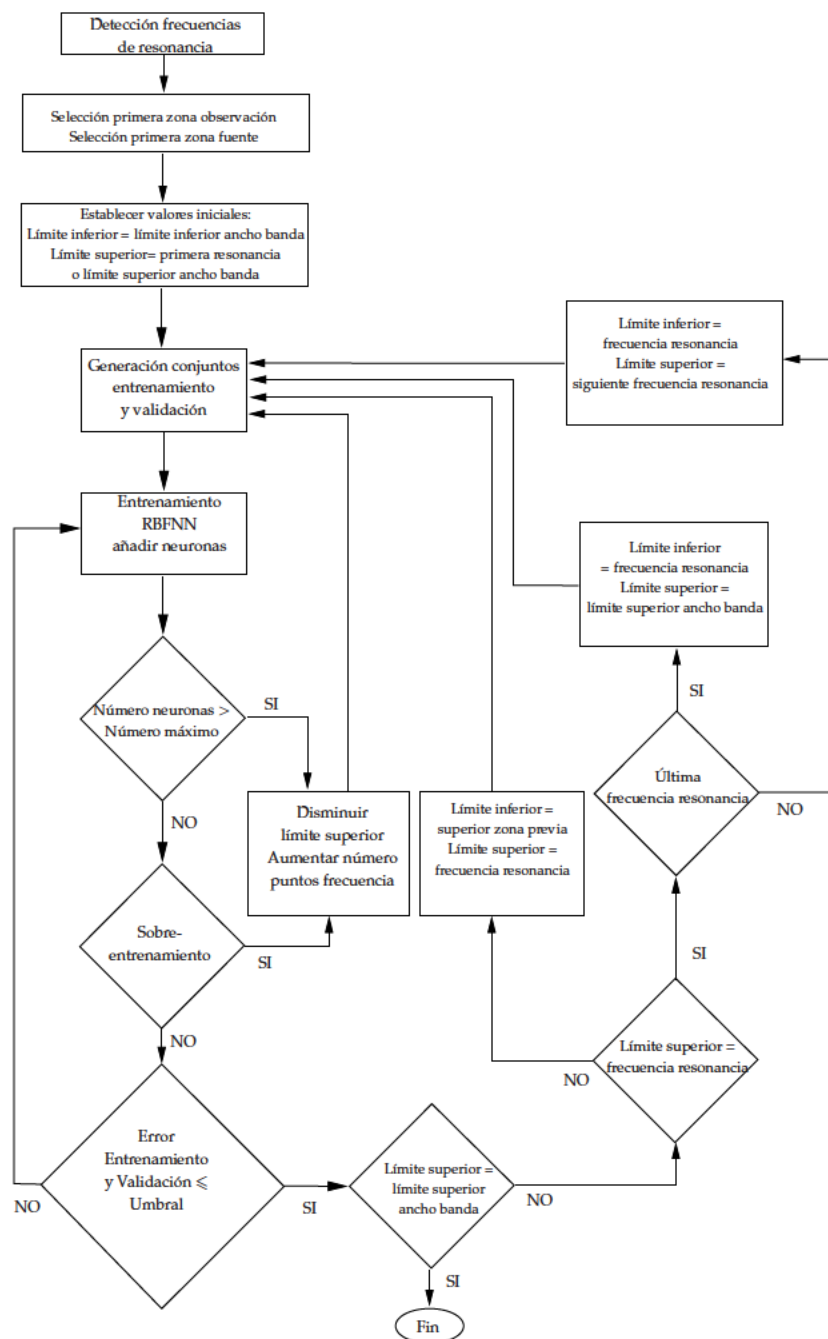


Figura 3.31: Diagrama de flujo del algoritmo de división en zonas de la coordenada frecuencia.

Para comprobar la validez del mecanismo de división, éste fue aplicado a la estructura dibujada en la Figura 3.7 en un ancho de banda entre 1 y 4 GHz. En la Tabla 3.4 pueden verse las

zonas observación generadas en el plano transversal. Así, la primera zona observación está compuesta por puntos observación cuyas coordenadas se sitúan en los siguientes límites, $x/a \in [x'/a - 0,0208, x'/a + 0,208]$ e $y/b \in [y'/b - 0,0208, y'/b + 0,208]$. La segunda zona observación, o primera zona intermedia, comienza en 0,0208 y tiene como límite superior 0,0417. En total se definen dos zonas intermedias. La división de los puntos observación se completa con una zona de puntos más alejados de la fuente. La división de puntos fuente se limita a aquellas fuentes cuyos puntos observación pertenecen a la primera zona observación. La Tabla 3.5 muestra las cuatro zonas en las que queda dividido el primer cuadrante del plano XY. La cavidad presenta una frecuencia de resonancia en 3 GHz, por tanto es necesaria una división frecuencial que tenga en cuenta este hecho. El ancho de banda entre el límite inferior de 1 GHz y la frecuencia de resonancia se divide en tres zonas tal como se observa en la Tabla 3.6. Además, se generan tres zonas frecuenciales entre la frecuencia de resonancia y el límite superior del ancho de banda de trabajo. Una zona frecuencial adicional es necesaria para aproximar las funciones de Green en la propia frecuencia de resonancia. El valor δ_f se define como un valor suficientemente pequeño para que el análisis electromagnético de los circuitos encapsulados tenga una precisión conveniente alrededor de la frecuencia de resonancia. Si $\delta_f = 0,01$ GHz entonces el análisis en torno a la frecuencia de resonancia tendrá que evitar aquellas frecuencias pertenecientes a $[2,99, 3,01]$ GHz. Este es un margen suficientemente pequeño para analizar correctamente cualquier circuito impreso en la estructura.

Tabla 3.4: Zonas observación generadas.

Zona	Límite inferior	Límite superior
1ª zona	Posición del punto fuente	0.0208
2ª zona	0.0208	0.0417
3ª zona	0.0417	0.1250
4ª zona	0.1250	1.0000

Tabla 3.5: Zonas fuente generadas para la primera zona observación.

Zona	Límite inferior	Límite superior
1ª zona	0.0833	0.5000
2ª zona	0.0208	0.0833
3ª zona	0.0104	0.0208
4ª zona	0.0000	0.0104

A la vista de las zonas generadas, son necesarias un total de 49 RBFNNs para aproximar las funciones de Green en todo el plano transversal para el ancho de banda deseado. Existen cuatro zonas fuente para la primera zona observación. Por tanto, la combinación de estas zonas con las zonas frecuenciales produce un total de 28 RBFNN distintas. Cada una de las dos zonas intermedias y la zona de puntos observación más alejados tienen asociada una zona de puntos fuente que cubre el primer cuadrante $0,0 \leq (x', y') \leq 0,50$. Así, la combinación de estas zonas espaciales genera tres conjuntos de entrenamiento y validación diferentes para cada zona frecuencial. De esta manera, se generan los conjuntos de datos para diseñar las 21 RBFNNs restantes.

Una vez generados todos los conjuntos de datos, el algoritmo OLS se encarga de seleccionar los

Tabla 3.6: Zonas frecuenciales generadas.

Zona	Límite inferior (GHz)	Límite superior (GHz)
1ª zona	1.00	2.50
2ª zona	2.50	2.90
3ª zona	2.90	$3.00 - \delta_f$
4ª zona	3.00	3.00
5ª zona	$3.00 + \delta_f$	3.10
6ª zona	3.10	3.50
7ª zona	3.50	4.00

centros de las gaussianas. Los pesos de la capa lineal son calculados mediante rutinas del paquete matemático LAPACK, basadas en la descomposición en vectores propios de la matriz de regresores final. La varianza debe ser fijada con anticipación al comienzo del algoritmo. Debido a que el rango de variación de las coordenadas espaciales es distinto al de la coordenada frecuencia, la varianza espacial es distinta al valor de varianza de la frecuencia. Así, la matriz diagonal de covarianza Σ^{-1} estará formada por los cuatro valores invertidos de las varianzas de las coordenadas espaciales $1/\sigma_1^2 = 1/\sigma_2^2 = 1/\sigma_3^2 = 1/\sigma_4^2 = 1/\sigma_s^2$ y la varianza invertida de la frecuencia $1/\sigma_5^2 = 1/\sigma_f^2$. Gracias a la asignación de un valor de varianza propio a cada tipo de coordenada se consiguen gaussianas mejor adaptadas a las variaciones de las funciones de Green en función de las variaciones de los datos de entrada. Se podría haber tomado la solución alternativa basada en la normalización de todos los datos de entrada a fin de conseguir un único rango de variación para todas las coordenadas. De este modo se podría utilizar un mismo valor de varianza para todas las coordenadas. Sin embargo, se ha optado por la diferenciación en la varianza, ya que las variaciones de las funciones de Green respecto a las variables espaciales son de diferente naturaleza que las variaciones debidas al cambio en la frecuencia. Además, la diferenciación permite de forma sencilla el desarrollo de algoritmos de cambio en la varianza, que como veremos en siguientes secciones, posibilitan una importante disminución en el nivel de error de aproximación.

Tal como se vio en la Sección 3.5 el algoritmo OLS es mucho más lento que los algoritmos no supervisados en la selección de los parámetros neuronales. Sin embargo, gracias a la supervisión en la elección de los centros de las gaussianas, el algoritmo OLS alcanza una precisión mayor que los algoritmos no supervisados. Para comprobar la capacidad de aproximación del algoritmo OLS y de un algoritmo no supervisado como el de agrupación en K-medias, se realizó un test con los puntos observación más cercanos a la fuente. Ambos algoritmos fueron empleados en la aproximación de las funciones de Green en la primera zona observación cuando los puntos fuente se encuentran en la primera zona fuente y para la zona frecuencial con límites 3,5 – 4 GHz. En este test numérico, los puntos observación se distribuyen en una rejilla formada por 49 puntos alrededor de cada punto fuente. Existen a su vez 49 puntos fuente que cubren de forma regular la primera zona fuente descrita en la Tabla 3.5. Finalmente, la zona frecuencial es muestreada con tres puntos. Así, el conjunto de entrenamiento estaba compuesto por 7203 elementos. Se eligió un conjunto de validación formado por puntos observación, fuente y frecuencia situados en posiciones intermedias respecto a los puntos de entrenamiento. Para examinar de forma firme la capacidad de generalización, la zona observación

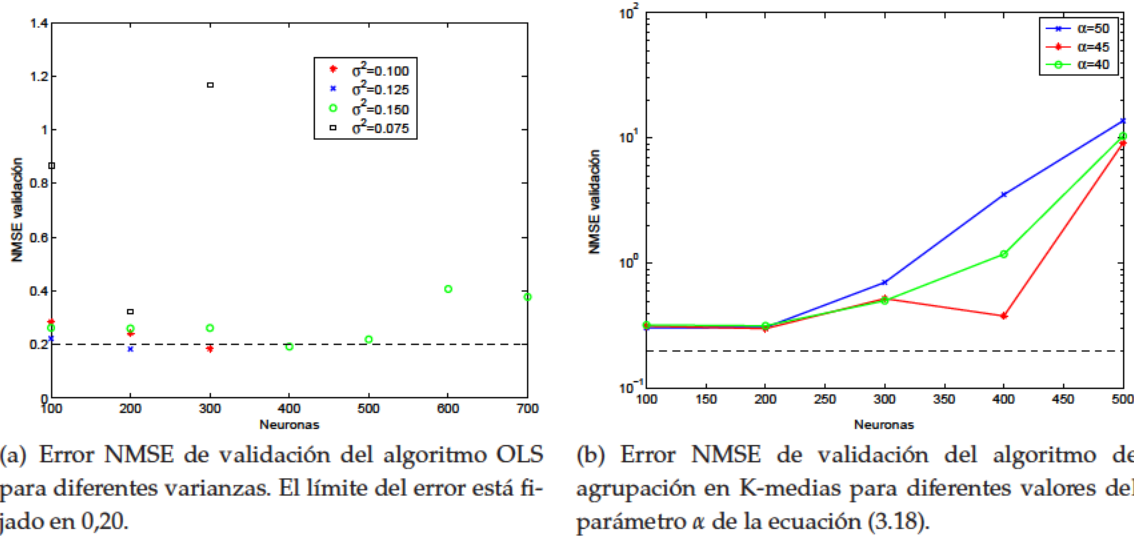


Figura 3.32: Comparativa del rendimiento entre los algoritmos OLS y K-medias para la primera zona observación.

fue muestreada con una rejilla de 9x9 puntos, más densa que la red de puntos observación de entrenamiento.

En las Figuras 3.32 se aprecia que el algoritmo OLS supera ampliamente el rendimiento del algoritmo de K-medias. Para ciertos valores de varianza el algoritmo supervisado es capaz de sobrepasar el error NMSE de validación máximo fijado. Merece destacar la dependencia de este algoritmo con la varianza. Si la varianza toma un valor demasiado pequeño ($\sigma^2 = 0,050$) para el problema dado el algoritmo diverge y se produce sobre-entrenamiento, de modo que el error de validación crece de forma abrupta. Si por el contrario la varianza toma un valor demasiado grande el algoritmo aprende de los datos de entrenamiento lentamente. Cuando la varianza es $\sigma^2 = 0,15$ el error de validación supera la barrera de 0,20 pero en esa situación el error NMSE de entrenamiento se encontraba ligeramente por encima del límite del error. Con la varianza de mayor valor existe sobre-entrenamiento pero la degradación de la aproximación es menor que en el caso de menor varianza. Este hecho es debido a que si la varianza es menor, las gaussianas poseerán una anchura menor, de forma que tendrán un valor nulo en prácticamente todo su ámbito de influencia. De esta manera, las gaussianas construidas con valores de varianza muy pequeños se aproximan a deltas de Dirac situadas en los puntos de entrenamiento. La capacidad de interpolación de este tipo de gaussianas es muy limitado. Así, la aproximación en el conjunto de entrenamiento será mejor conforme más neuronas se añadan a la RBFNN, pero el error de generalización será elevado. Para que el algoritmo OLS alcance toda su habilidad de aproximación es imprescindible un valor de varianza adecuado, como $\sigma^2 = 0,10$ o $\sigma^2 = 0,125$.

El algoritmo de K-medias ofrece un rendimiento muy bajo en este problema. Su capacidad de generalización se revisó para distintos valores del parámetro α que controla la varianza de las gaussianas según la ecuación (3.18). En la Figura 3.32(b) se muestran los valores de este parámetro que mejores resultados ofrecieron. Sin embargo, todas las ejecuciones conducen a un fuerte sobre-entrenamiento. El mejor valor de validación se situó en torno a un valor de NMSE de 0,30. El al-

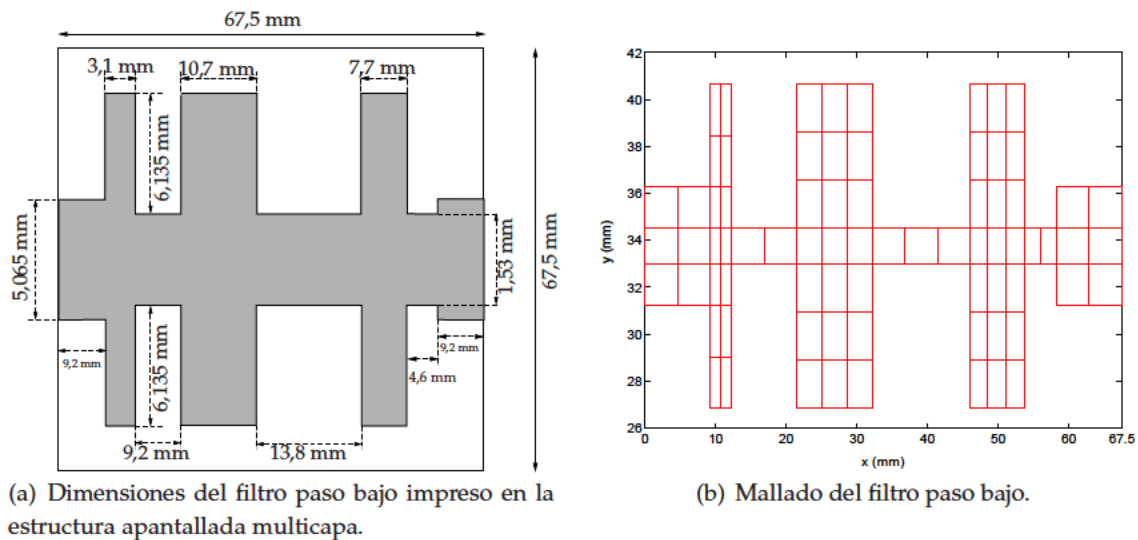


Figura 3.33: Estructura del filtro paso bajo impreso en el plano transversal de la cavidad apantallada multicapa de la Figura 3.7.

goritmo no supervisado de agrupación de centros no aporta ninguna ventaja respecto al OLS ya que los datos de entrada no se aglutinan en grupos, sino que se encuentran situados en una red regular. El algoritmo OLS depende fuertemente de la varianza, que debe ser fijada en ocasiones de forma heurística. No obstante, el algoritmo de K-medias también depende de diversos parámetros que deben ser fijados por el usuario. Así, la varianza dependerá de parámetros como α o como el número de centros para realizar la media, si se emplea la ecuación (3.18). Otras expresiones para calcular la varianza de las neuronas en el algoritmo de K-medias dependen de parámetros heurísticos, [Schwenker et al., 2001]. En la primera versión del algoritmo de división, la varianza se fija de forma heurística por parte del usuario. La varianza seleccionada debe poseer un valor dentro de los márgenes que aseguran una correcta aproximación de las funciones de Green. Además este valor de varianza se aplica a todas las zonas observación.

Las RBFNNs entrenadas fueron integradas en una formulación de la IE para formar la herramienta de análisis neuronal. En lugar de recurrir a la evaluación de las funciones de Green, la IE resuelta con el MoM obtiene los datos requeridos desde la salida de las RBFNNs. Para cada dato compuesto por un punto fuente, un punto observación y una frecuencia determinada es imprescindible seleccionar a qué parte del espacio de entrada pertenece. Toda región del espacio tiene una RBFNN asociada que aproxima las funciones de Green de forma eficiente. La RBFNN asociada al dato de entrada calcula las partes reales e imaginarias de las funciones de Green de forma veloz debido al reducido número de neuronas que compone dicha RBFNN. Gracias al mecanismo de división en zonas, cada vez que se necesita un dato de salida se recurre a una RBFNN de un tamaño relativamente pequeño, en lugar de emplear una red neuronal muy grande y por tanto, más lenta. La completa aproximación de las funciones de Green en todo el plano transversal y en todo el ancho de banda de trabajo posibilita que cualquier circuito impreso pueda ser analizado de forma fiable y rápida. Para validar la técnica de análisis desarrollada se simuló la respuesta del filtro paso bajo de la Figura 3.33.

Como se observa en la Figura 3.34 la respuesta producida por el método de análisis neuronal sigue con fidelidad la respuesta obtenida con la IE que emplea la evaluación directa de las funciones de Green. En todos los análisis de la tesis se ha empleado la versión de Garlekin del MoM con funciones de base de tipo tejado ("rooftop"). La excitación en los puertos es de tipo "delta-gap". Para validar el método desarrollado se muestran asimismo los valores medidos extraídos de [Eleftheriades et al., 1995]. Como se aprecia en la Figura 3.34 los valores simulados exhiben gran similitud con los valores medidos.

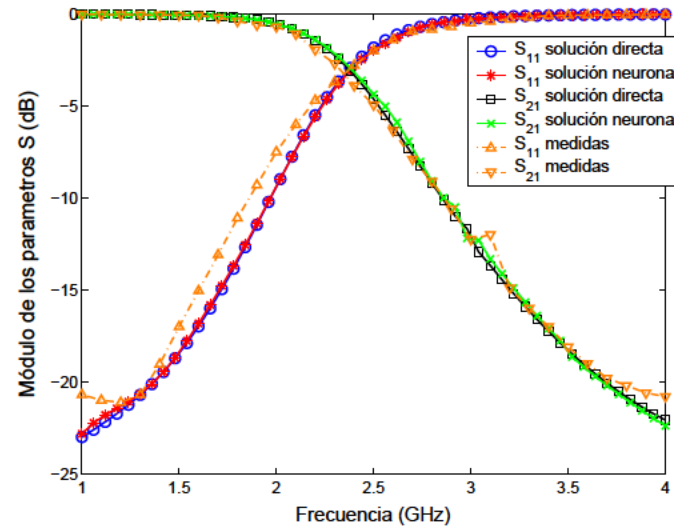


Figura 3.34: Parámetros S del filtro paso bajo de la Figura 3.33(a).

La razón para utilizar las RBFNNs en lugar de la evaluación numérica de las funciones de Green radica en la gran rapidez con que las redes neuronales calculan los valores de salida. En la Tabla 3.7 se aprecia la gran ganancia temporal conseguida con el método neuronal. El tiempo por punto en frecuencia cuando se combinan las RBFNNs con la IE es de tan solo 4,6 segundos. En cambio, si las funciones de Green exactas son empleadas el tiempo requerido para calcular los parámetros de "scattering", o parámetros S, por punto en frecuencia es de 17,7 minutos. El método neuronal es por tanto en torno a 200 veces más rápido que la solución directa empleada.

En la Tabla 3.1 se muestra el tiempo de cálculo de las funciones de Green y el de las salidas de la RBFNN entrenada para los conjuntos de entrenamiento y validación de la Figura 3.16. En ese caso, la ganancia temporal no es tan grande como en el análisis del filtro paso bajo. Este hecho es debido a que los conjuntos de entrenamiento y test están formados por una serie de puntos observación distribuidos de forma uniforme para cada punto fuente. De este modo, la evaluación de las funciones de Green en la Tabla 3.1 se realiza para un número de puntos cerca de la fuente similar al número de puntos alejados de la fuente. Sin embargo, la estimación numérica de las funciones de Green es más costosa conforme más se acerca el punto observación al punto fuente, incluso si se emplea la rápida suma de imágenes espaciales. Cuando la IE es resuelta con el MoM, el número de puntos integración necesario para calcular las interacciones entre celdas próximas es mayor que el número requerido cuando las celdas se encuentran alejadas. De hecho, cuando se calcula la interacción de una celda consigo misma el número de puntos de integración es máximo. De este modo, la mayor parte de las

evaluaciones de las funciones de Green se realizan en puntos observación situados en las cercanías de la fuente. En estos puntos observación la ganancia temporal obtenida por las RBFNNs es mayor, provocando que la ganancia temporal del análisis mostrado en la Tabla 3.7 sea mayor que la señalada en la Tabla 3.1.

El tiempo requerido para generar los conjuntos de entrenamiento y validación y para entrenar las RBFNNs necesarias fue 4 horas y 37 minutos [Garcia et al., 2005a]. El proceso de diseño de las RBFNNs solo debe ser realizado una única vez para una estructura y ancho de banda determinados. Por tanto, el periodo invertido en el diseño de las RBFNNs es un tiempo inicial que se convierte en insignificante a medida que se simulan circuitos en la cavidad. Por ejemplo, en el caso de aplicar un proceso de optimización sobre el circuito impreso, cada ejecución del análisis neuronal costará tan solo unos minutos en lugar de horas del método directo. La ganancia temporal conseguida con el método neuronal depende directamente del número de neuronas que componen las RBFNNs. En la estructura de la Figura 3.7 las RBFNNs entrenadas poseen un tamaño medio de 511 neuronas con una desviación estándar de 110 neuronas. El error NMSE máximo que deben alcanzar las RBFNNs entrenadas es de 0,20.

Tabla 3.7: Comparación de los resultados temporales de la solución directa y el método neuronal para el cálculo de los parámetros S en 51 puntos de frecuencia.

Tiempo de diseño	Solución directa	Solución neuronal
4 horas 37 minutos	15 horas 2 minutos	237.7 segundos

3.8. Algoritmo de división del espacio de entrada con varianza adaptativa

El algoritmo de división genera múltiples zonas de forma que se pueden construir RBFNNs especializadas en cada una de las regiones en las que se divide el espacio de entrada. Esta característica produce redes neuronales eficientes en la aproximación de las funciones de Green. La división del espacio tiene en cuenta la localización de los puntos observación y fuente para conseguir una completa representación de las funciones de Green. Gracias a esta división dirigida, las RBFNNs entrenadas con el algoritmo OLS alcanzan un nivel de error NMSE máximo de 0,20 en el ejemplo de la Sección anterior. La precisión alcanzada permite que el método neuronal fruto de la utilización de las RBFNNs en la IE sea capaz de analizar con exactitud y gran rapidez el filtro paso bajo de la Figura 3.33 [Garcia et al., 2005a]. Este tipo de filtros no presentan acoplos críticos entre las tiras metálicas. Dichos acoplos críticos entre los elementos que componen el filtro impreso son difíciles de caracterizar. Por tanto, si la estructura bajo análisis presenta pequeñas distancias de acoplo entre las distintas tiras metálicas es necesario que las RBFNNs representen con gran precisión las funciones de Green para expresar correctamente los acoplos existentes.

En la primera versión del algoritmo, la división en zonas fuente se limitaba a la primera zona observación. En esta nueva versión, para expresar de modo más completo el comportamiento de las funciones de Green, la división espacial de las fuentes es ampliada a las zonas observación intermedias. La división de las fuentes se sigue realizando en esta segunda versión con puntos observación pertenecientes a la primera zona observación. La zona de puntos más cercanos a la fuente es aquella

de más difícil aproximación. Así, el número de zonas fuente será lo suficientemente abundante para asegurar también un correcto entrenamiento para todas las zonas intermedias. Todas las zonas de observación intermedias y la primera zona observación tendrán asignadas las mismas zonas fuente. Al igual que en la primera versión del algoritmo, la zona de puntos observación más alejados de la fuente tiene asociada una zona con puntos fuente que cubren totalmente el primer cuadrante. En la Figura 3.35 se observa el diagrama de flujo del proceso de división de las zonas observación que se ha aplicado en la segunda versión del algoritmo de división [Garcia et al., 2006b].

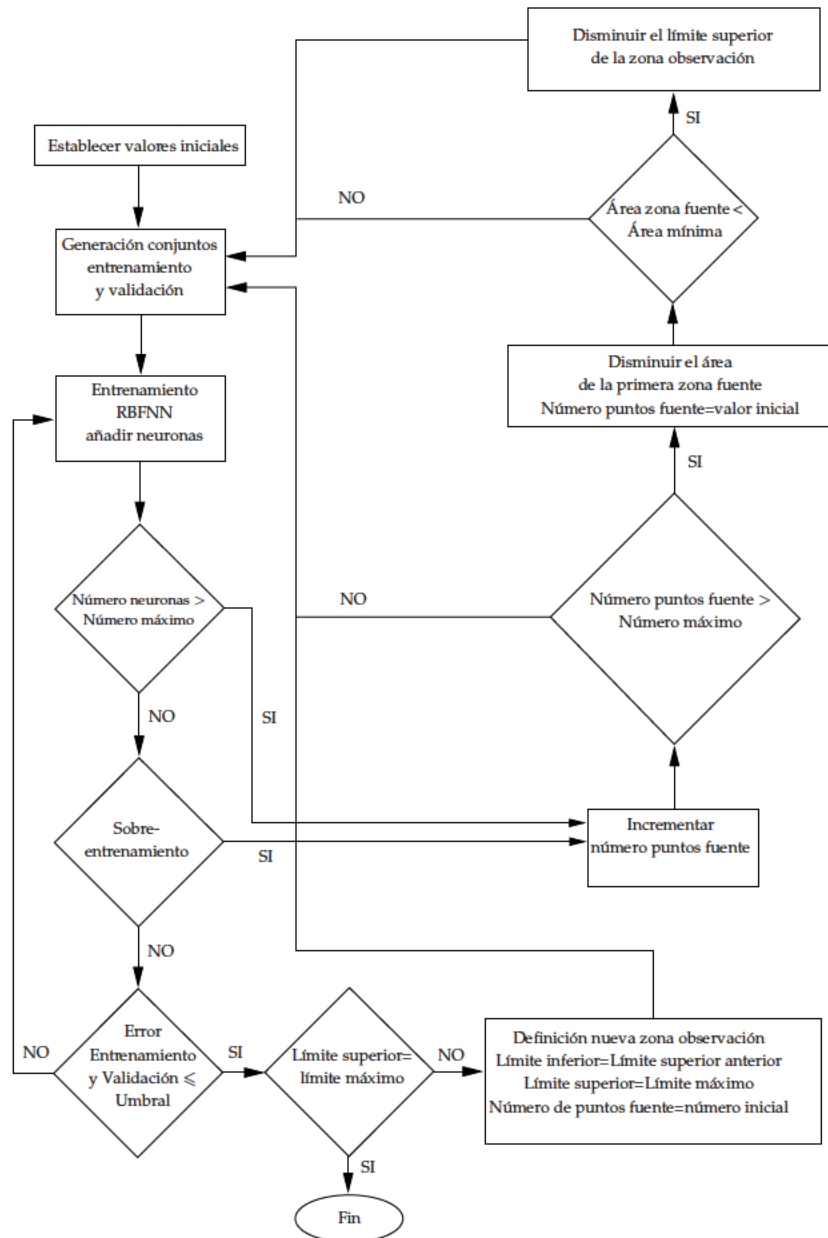


Figura 3.35: Diagrama de flujo del algoritmo de división en zonas de los puntos observación.

El algoritmo descrito en el diagrama de flujo de la Figura 3.35 permite que las zonas observación sean delimitadas de forma automática. La única intervención del diseñador se produce en el inicio del algoritmo con la fijación de ciertos parámetros. Entre los valores que deben ser fijados podemos

citar la varianza, los límites mínimos de la primera zona fuente o el número de puntos observación de las zonas que se deben definir. La frecuencia se fija al valor máximo del ancho de banda de trabajo por las razones explicadas en la Sección 3.7. Una vez especificados los parámetros fundamentales del proceso de división comenzará el entrenamiento. Si se detecta una situación de sobre-entrenamiento o se supera el número máximo de neuronas se debe aumentar la densidad de puntos en el conjunto de entrenamiento. De esta forma, se aumentará la capacidad de interpolación y se podrá incrementar el número de neuronas sin que el error de generalización aumente. El aumento de la densidad de puntos se lleva a cabo en primera instancia mediante el incremento del número de puntos fuente. Si debido a limitaciones en la memoria no se puede aumentar el tamaño del conjunto de entrenamiento, se procede a la reducción de la primera zona fuente. En el caso de que las anteriores operaciones no tengan éxito, se recurre a la disminución del área de la zona observación. Tras la selección de una zona observación determinada empieza la definición de la siguiente zona observación. El límite superior de la posible nueva zona es el límite máximo de las zonas intermedias mientras que su límite inferior es el límite superior de la zona anterior. El algoritmo finaliza cuando el límite superior de una zona intermedia coincide con el límite máximo fijado al principio del proceso.

Al igual que el mecanismo de división de zonas observación, el proceso de división de zonas fuente comienza con la fijación de ciertos parámetros. Entre los parámetros a fijar encontramos el número máximo de neuronas, la cantidad máxima de puntos fuente o el límite mínimo de la zona fuente. El valor de varianza utilizado es el mismo que el empleado en el proceso de división en zonas observación. De forma similar a la división de zonas observación, el proceso de selección automática de zonas fuente debe detectar si aparece una situación de sobre-entrenamiento. Como se aprecia en el diagrama de flujo de la Figura 3.36, en caso de que el error de validación aumente de forma abrupta se procede a incrementar el número de puntos fuente. Si el aumento de la densidad de puntos fuente no produce una aproximación apropiada se debe reducir el tamaño de la zona fuente. La disminución del límite inferior de la zona fuente conlleva un aumento de la densidad de puntos de forma que la información que dispone la RBFNN para aproximar las funciones de Green se incrementa. Cuando se supera el límite máximo de error NMSE de entrenamiento y validación se crea una nueva zona fuente. Siguiendo este proceso, se generan sucesivamente una serie de zonas fuente en forma de "L" que cubren el primer cuadrante del plano transversal. Hay que tener en cuenta que la primera zona fuente en forma de cuadrado es proporcionada por el proceso de selección de zonas observación descrito en el diagrama de flujo mostrado en la Figura 3.35. Tanto en el algoritmo de división de zonas observación como en el de zonas fuente se deben establecer unos límites máximos de puntos fuente y observación apropiados para la tarea de aproximación. Dichos límites deben garantizar la generación de conjuntos de entrenamiento con un número de muestras suficiente para evitar el problema del sobre-entrenamiento. Al mismo tiempo deben mantener un tamaño del conjunto de entrenamiento que evite la aparición de problemas de computación. Si se fija un tamaño de entrenamiento muy reducido el algoritmo dividirá el espacio de entrada en muchas zonas. El algoritmo de división es por tanto flexible para generar un número de zonas adecuado. Si por el contrario, el tamaño de entrenamiento es elevado, se proveerán un número reducido de zonas. Sin embargo, un tamaño máximo de entrenamiento relativamente reducido es más adecuado. Por un lado, permite la creación de RBFNNs que operan sobre conjuntos de entrenamiento pequeños. Así, poseerán una cantidad reducida de neuronas acelerando el análisis neuronal. Por otro lado, si el tamaño máximo de entrenamiento es elevado pueden aparecer problemas de computación debido a la saturación de

la memoria. Aunque la generación de muchas zonas puede aumentar el tiempo dedicado a esta parte del diseño de las RBFNNs, hay que tener en cuenta que es un periodo inicial. Como se ha visto en el filtro paso bajo de de la Figura 3.33 la ganancia temporal convierte en muy rentable el tiempo invertido en el diseño y entrenamiento de las RBFNNs.

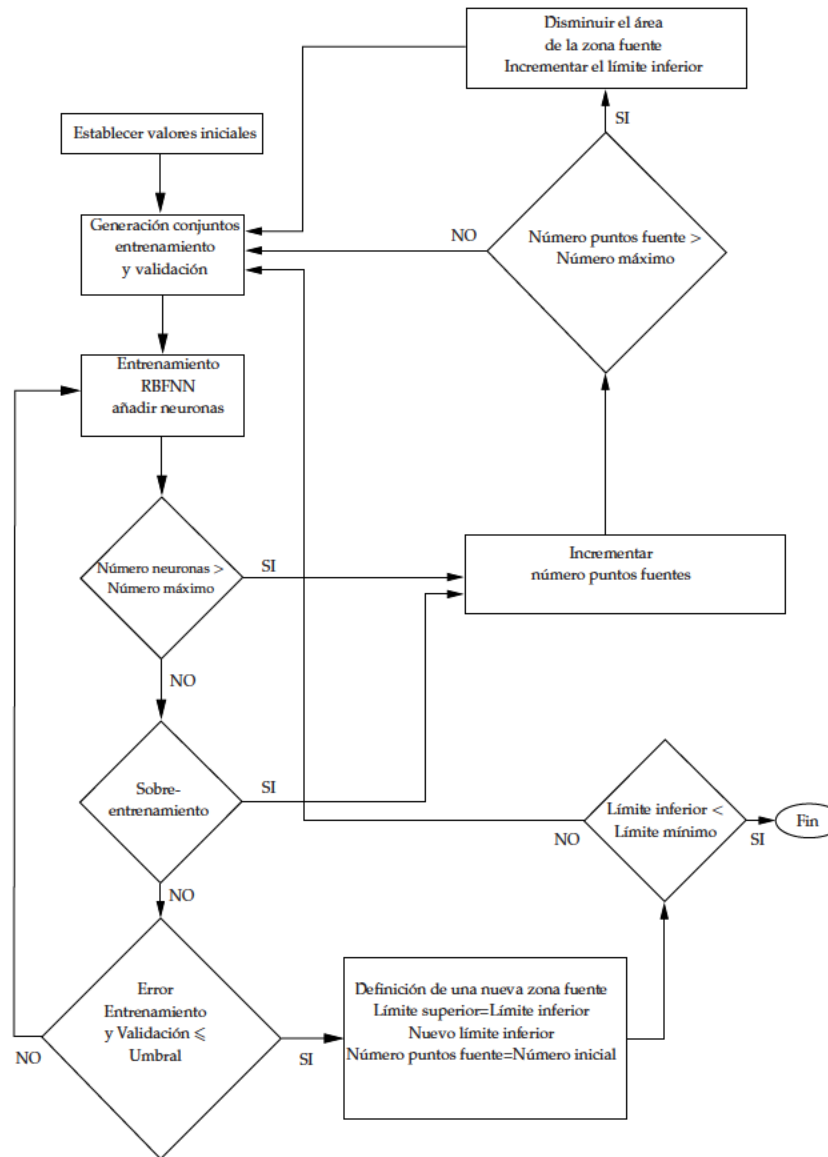


Figura 3.36: Diagrama de flujo del algoritmo de división en zonas de los puntos fuente.

Los algoritmos automáticos descritos en los diagramas de flujo de las Figuras 3.35, 3.36 y 3.31 fueron aplicados en la estructura encapsulada de la Figura 3.37, en un ancho de banda de trabajo de 9 – 11 GHz. La extensión de la división en zonas fuente a las zonas intermedias de puntos observación permite la consecución de conjuntos de entrenamiento de menor tamaño, que además proporcionan menores niveles de error. En [Garcia et al., 2006b] se presenta un test numérico que muestra la utilidad de la extensión realizada. La división de zonas fuente fue aplicada a una zona intermedia cuyo límite superior es de 0,0208 y límite inferior 0,0104. Si no se emplea ninguna división sobre las fuentes, el error NMSE mínimo logrado es de 0,48. En este ejemplo el tamaño máximo del conjunto

de entrenamiento es fijado en 4059 muestras debido a limitaciones en la memoria (con este tamaño máximo y tres puntos en frecuencia para una zona frecuencial, el tamaño total del conjunto de entrenamiento para una RBFNN sería superior a 12000 muestras). Si no se realiza ninguna división en la región del espacio de entrada correspondiente a los puntos fuente, no se podrán alcanzar niveles menores de error. El algoritmo de división de zonas fuente consigue que el error NMSE caiga por debajo de 0,20. Para alcanzar este nivel de error son necesarias diversas zonas fuente cuyo tamaño medio es de 2662 muestras. Por tanto, gracias a la extensión del algoritmo desarrollado en la Sección 3.7, se consigue una aproximación más precisa con conjuntos de entrenamiento de menor tamaño.

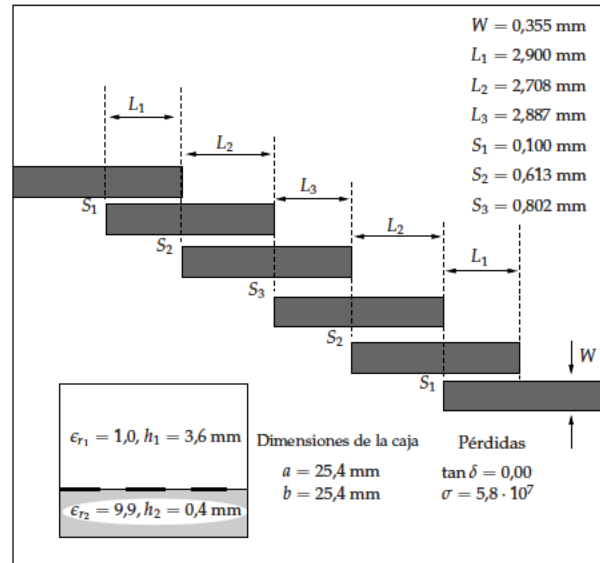


Figura 3.37: Filtro paso banda de cuatro polos basado en líneas acopladas. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).

Los algoritmos de división del espacio de entrada producen las zonas observación y fuente mostradas en las Tablas 3.8 y 3.9 respectivamente. El umbral de error NMSE máximo se fijó en 0,20 para todas las RBFNNs entrenadas. En primer lugar el algoritmo de división de los puntos observación generó cuatro zonas. En estas zonas se incluye la zona de puntos más cercanos a la fuente, dos zonas intermedias y la zona de puntos más alejados de la fuente. Empleando la primera zona observación se definieron seis zonas de puntos fuente. El valor de varianza se fijó en $\sigma_s^2 = 0,050$ para todas las zonas observación. Este valor se fijó de forma heurística, en base a pruebas realizadas previamente al comienzo del algoritmo de división. Para representar las variaciones respecto de la frecuencia fue necesaria únicamente una zona frecuencial que cubre todo el ancho de banda. El algoritmo de división de la coordenada frecuencia halla esta única zona frecuencia de forma simple ya que no existe ninguna frecuencia de resonancia en el ancho de banda de trabajo. Además, el muestreo de dicha zona se limitó a tres puntos (9 GHz, 10 GHz, 11 GHz). La varianza frecuencial tomó un valor intermedio de $\sigma_f^2 = 10,0$ que garantiza una correcta aproximación.

Con el objetivo de comprobar la habilidad del método neuronal en el análisis de filtros que involucran acoplos críticos, se simuló la respuesta de los circuitos dibujados en las Figuras 3.37 y 3.38. En las Figuras 3.39 y 3.40 se observan los parámetros S calculados mediante la solución directa

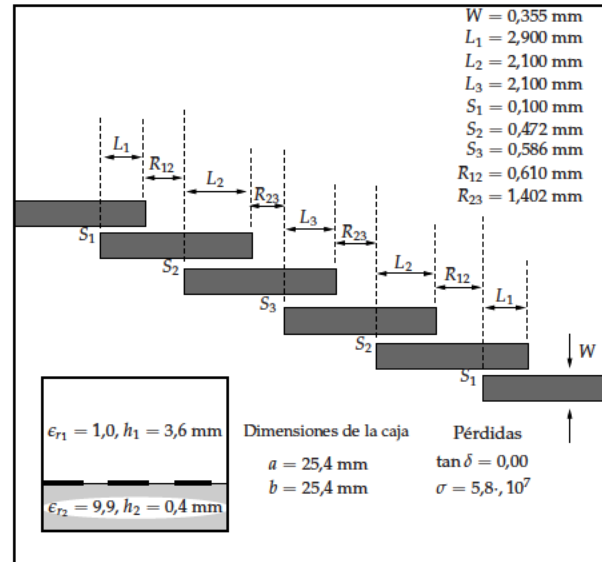


Figura 3.38: Filtro paso banda de cuatro polos. Las líneas acopladas se conectan con una longitud extra de línea de transmisión. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).

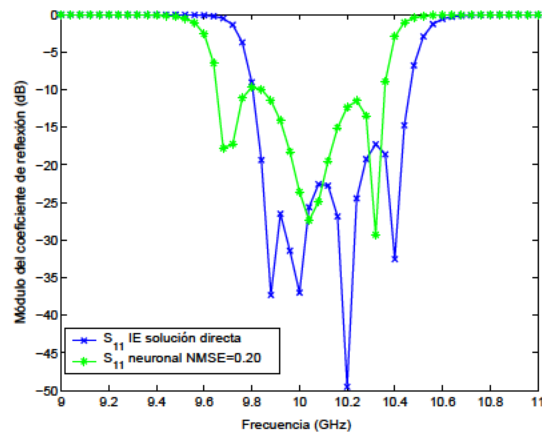
Tabla 3.8: Zonas observación generadas por el algoritmo de división con varianza fija.

Zona	Límite inferior	Límite superior
1ª zona	Posición del punto fuente	0.02080
2ª zona	0.02080	0.06250
3ª zona	0.06125	0.12500
3ª zona	0.01250	0.10000

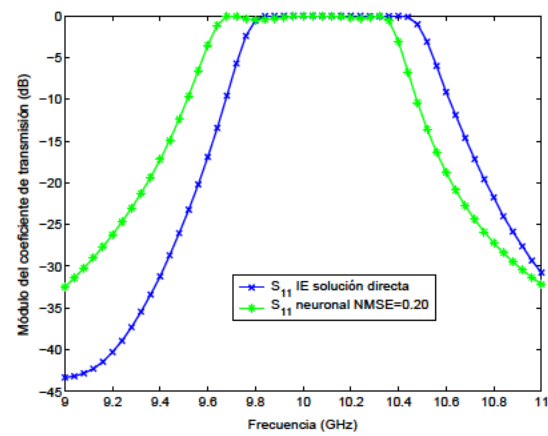
que hace uso de las funciones de Green evaluadas numéricamente y la solución correspondiente al método neuronal. En ambos casos se empleó el mismo mallado dibujado en las Figuras 3.41 durante la resolución de la IE con el MoM. EL nivel de error NMSE de 0,20 no es suficiente para alcanzar la precisión requerida en el análisis de este tipo de filtros. La respuesta proporcionada por el método neuronal se encuentra desplazada en frecuencia de forma significativa. Además, la característica de reflexión en la banda de paso para ambos filtros no coincide con el nivel marcado en la respuesta

Tabla 3.9: Zonas fuente generadas por el algoritmo de división con varianza fija.

Zona	Límite inferior	Límite superior
1ª zona	0.3000	0.5000
2ª zona	0.2000	0.3000
3ª zona	0.1250	0.2000
4ª zona	0.0688	0.1250
5ª zona	0.0266	0.0688
6ª zona	0.0000	0.0266

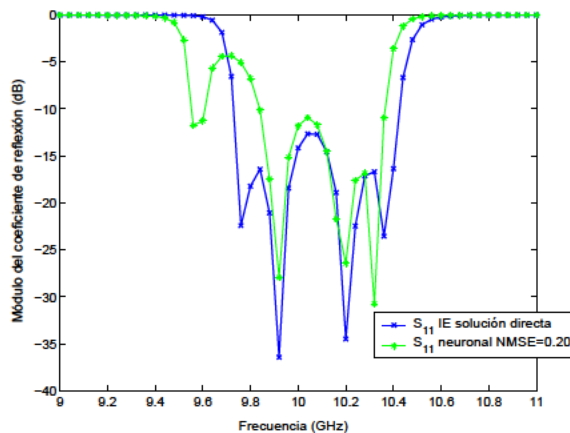


(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

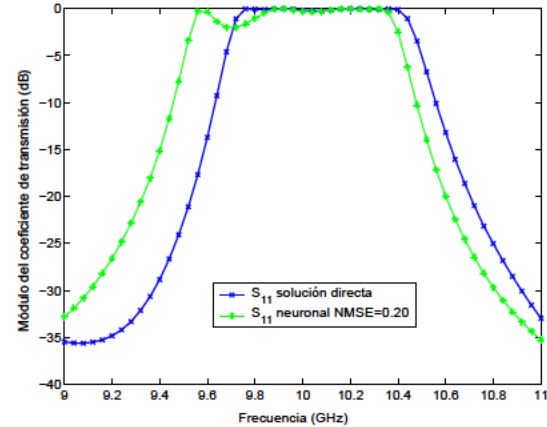


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

Figura 3.39: Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,20.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

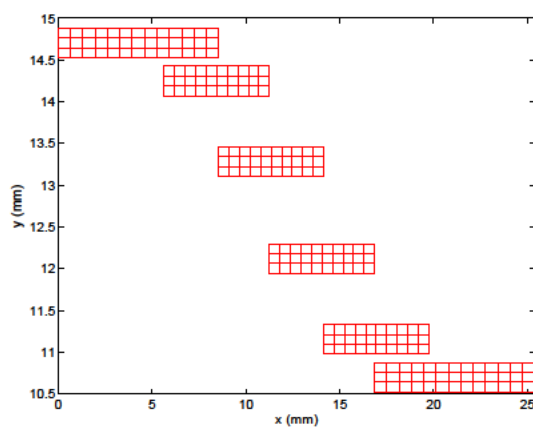


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

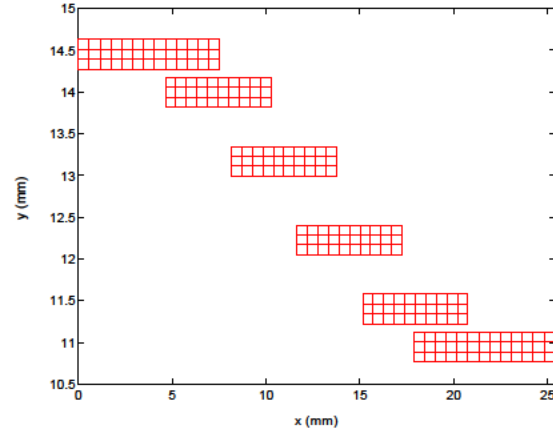
Figura 3.40: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,20.

directa. Por lo tanto, es necesario aumentar la calidad de la aproximación de las funciones de Green por parte de las RBFNNs para garantizar la precisión en el análisis de circuitos complejos.

En las Figuras 3.18, 3.19, 3.21, 3.22, 3.32(a) se puede comprobar la importancia del valor de varianza en el rendimiento de la RBFNN. En el algoritmo OLS los valores de la diagonal de la matriz de covarianza deben ser establecidos con anterioridad al inicio de la selección de los centros. En los anteriores algoritmos de división del espacio de entrada, la varianza es común a todas las RBFNNs diseñadas. Aunque existe un rango de varianzas que permiten un entrenamiento correcto, una elección inapropiada de valores en la matriz de covarianza puede incrementar los niveles de error de entrenamiento y validación tal como se muestra en [Garcia et al., 2004b] y [Garcia et al., 2004c]. Cada RBFNN está especializada en la aproximación de las funciones de Green de una combinación de



(a) Mallado del filtro paso banda de cuarto orden mostrado en la Figura 3.37.



(b) Mallado del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

Figura 3.41: Mallados de los filtros de cuatro polos mostrados en las Figuras 3.37 y 3.38.

una zona observación, una zona fuente y una zona frecuencia determinadas. Aquellas RBFNNs cuya zona observación se sitúe cerca de la fuente deberán aproximar funciones de rápido crecimiento. En cambio, las RBFNNs en las que los puntos observación se encuentran lejos de los puntos fuente aproximarán funciones de crecimiento lento. Las varianzas de la matriz de covarianza de las funciones gaussianas determinan si las salidas de la RBFNN son funciones de variación lenta o rápida. Si los valores de varianza son pequeños, las gaussianas crecen rápidamente en torno al centro. Si por el contrario los valores de varianza son elevados, las gaussianas tendrán una mayor “anchura” en torno a los centros, y por tanto los valores de salida mostrarán mayor suavidad. Así, si los valores comunes de varianza toman un valor demasiado pequeño, las zonas observación cercanas a la fuente serán aproximadas con mayor precisión que aquellas zonas alejadas de la fuente. Si por el contrario, las varianzas se fijan a un valor grande, dentro del rango posible de valores, entonces la aproximación cerca de la fuente se degradará en beneficio de la aproximación lejos de la fuente. Este hecho puede apreciarse en las Figuras 3.23 donde se muestra la aproximación de diversas rectas de puntos observación para distintas fuentes situadas en la cavidad de la Figura 3.7.

La importancia de la división del espacio de entrada y de la asignación de un valor de varianza correcto a cada una de las zonas queda patente en [Soliman et al., 2004]. En ese trabajo las RBFNNs aproximan los coeficientes de la matriz de acoplos del Método de los momentos. Para realizar tal aproximación se emplean dos RBFNNs, una para los términos Z_{xx} y otra para los términos Z_{xy} . No obstante, cuando la distancia entre celdas es pequeña, se utiliza la solución directa del MoM, de forma que no se emplea ninguna RBFNN. Este hecho es provocado porque las RBFNNs utilizadas no son capaces de aproximar las rápidas oscilaciones cerca de la fuente. Se produce por ello una merma en el rendimiento del método. En [Soliman et al., 2004], se usan las funciones provistas en *MATLAB*® para entrenar una RBFNN. Los autores exponen que una solución al problema radica en el aumento del número de puntos de entrenamiento mediante un muestreo no uniforme, más denso conforme menor es la distancia a la fuente. Esta solución no se implementa ya que el muestreo no uniforme es imposible en *MATLAB*®. Por tanto, la solución adoptada en [Soliman et al., 2004] consiste en emplear el MoM original en la zona de puntos más cercanos a la fuente. Sin embargo,

el aumento de la densidad de puntos cerca de la fuente se puede conseguir mediante la adecuada división del espacio de entrada. Además, si en las zonas cercanas a la fuente se utilizan menores valores de varianza, es posible la aproximación eficiente en todo el espacio. En *MATLAB*® no se puede fijar un valor de varianza diferente para cada coordenada del dato de entrada, pero sí es factible la alteración de la distribución de los datos de entrada de entrenamiento. Mediante esta transformación se obtiene un resultado similar al cambio de varianza. Al no emplear ninguna estrategia de división del espacio de entrada ni ningún mecanismo de asignación de varianza, el método propuesto en [Soliman et al., 2004] es sub-óptimo. En [Garcia et al., 2006b] estos problemas se solventan mediante la combinación del procedimiento de división en zonas del espacio de entrada con el mecanismo de búsqueda de varianza apropiada que se explica a continuación.

Para mejorar la capacidad de aproximación de las RBFNNs en el problema del cálculo de las funciones de Green es necesario que cada zona observación cuente con una serie de varianzas propias. Una vez fijados los valores de varianza para cada zona observación, éstos pueden conservarse para las zonas fuente y frecuencia, ya que el crecimiento de las funciones de Green depende en gran medida de la distancia entre punto observación y fuente. Las mayores dificultades en la aproximación de las funciones de Green se producen en las cercanías de la fuente. En la zona más cercana a la fuente será necesario un valor menor de varianza para expresar las rápidas variaciones de las funciones de Green. Sin embargo, los valores de varianza no pueden ser disminuidos indefinidamente para una distancia determinada entre puntos observación y fuente. Si dichos valores disminuyen por debajo de un valor mínimo, la RBFNN pierde la capacidad de generalización de forma que se produce sobre-entrenamiento. Este hecho negativo se debe a que las funciones gaussianas se convierten prácticamente en funciones delta de Dirac. Cada gaussiana tomará, de esta forma, un valor casi nulo para aquellos datos de entrada distintos de su centro. En esta situación cada neurona memoriza un dato de entrenamiento de modo que el error de entrenamiento es pequeño. Sin embargo, no es capaz de realizar una interpolación correcta cuando se presenta un nuevo dato. En este caso, la solución para mejorar la precisión consiste en aumentar el número de puntos de entrenamiento. Así, las funciones de Green son expresadas por el conjunto de entrenamiento de forma más completa. Además, con esta operación se disminuye la distancia entre puntos, de modo que se posibilita el uso de menores valores de varianza, más adecuados cuando la función presenta un rápido crecimiento.

Para solucionar estos inconvenientes se ha desarrollado un mecanismo que busca de forma adaptativa los valores de varianza apropiados para cada zona observación. Por tanto, la búsqueda de la mejor varianza se debe aplicar durante el algoritmo de división de zonas observación. Ahora, este algoritmo definirá las zonas observación, la primera zona fuente y los valores de la matriz de covarianza de cada zona observación. Una vez seleccionados los valores de varianza se mantendrán fijos para todas las zonas fuente. En esta fase del algoritmo de división en zonas, la frecuencia toma un valor constante e igual a la mayor frecuencia del ancho de banda. Por lo tanto, el algoritmo ajustará los valores de varianza espacial $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2 = \sigma_s^2$, mientras que la varianza frecuencial, σ_f^2 será calculada durante la ejecución del algoritmo de división de zonas frecuencia. En la Figura 3.42 se muestra el diagrama de flujo del algoritmo de selección de zonas observación combinado con el mecanismo de varianza adaptativa. Al igual que en el algoritmo original, el proceso de división comienza con el establecimiento de una serie de valores iniciales. Ciertos valores corresponden a parámetros que permanecen fijos durante todo el proceso. Podemos citar el límite máximo de las zonas

observación o el límite mínimo de la primera zona fuente. Otros valores son alterados por el algoritmo, como por ejemplo el tamaño inicial de la primera zona fuente o el valor de varianza σ_s^2 inicial. Si el umbral máximo de error NMSE de entrenamiento y validación no es alcanzado, la varianza espacial es disminuida en cada paso. Si el error en la aproximación no desciende hasta un valor por debajo del umbral, la varianza sigue reduciéndose hasta que se entrene la RBFNN de forma exitosa o hasta que se alcance el valor de varianza mínimo. Si se produce esta última situación, comienza una nueva fase en la que se incrementa la varianza espacial. En caso de no encontrar un valor de varianza óptimo por medio de las fases alternativas de decremento e incremento, se procede a operar sobre la densidad de puntos fuente. Para conseguir dicho aumento en la densidad de puntos, bien se aumenta el número de puntos o bien se disminuye el área de la zona fuente. Los dos mecanismos de aumento de la densidad de puntos se exploran de forma secuencial como se observa en el diagrama de flujo de la Figura 3.42. De esta forma, se pueden utilizar valores de varianza menores sin pérdida de la capacidad de generalización.

El nuevo algoritmo de división de zonas observación combinado con la estrategia de búsqueda de la mejor varianza fue aplicado a la estructura de la Figura 3.37. Asimismo, los algoritmos de división de zonas fuente y zonas frecuenciales fueron aplicados para completar la división del espacio de entrada. El nuevo mecanismo de búsqueda de la varianza posibilita el descenso del error NMSE en la primera zona observación hasta un valor de 0,10. Se ha permitido que los niveles de error NMSE en aquellas zonas distintas de la primera zona observación, sean menores que el umbral de error igual a 0,10 impuesto a todas las zonas en [Garcia et al., 2006b]. Este límite estaba determinado por el error alcanzado en la zona observación más cercana a la fuente. Sin embargo, lejos de la fuente la precisión puede aumentar debido a la ausencia de rápidas variaciones en las funciones de Green. De esta forma, se pueden establecer umbrales de error cada vez menores conforme nos alejamos de la fuente. Por ello, aunque la división en zonas es igual a la realizada en [Garcia et al., 2006b], las RBFNNs han sido re-entrenadas consiguiendo un cierto aumento de la precisión en aquellas zonas observación alejadas de la fuente.

Para alcanzar los distintos umbrales de error, el algoritmo de división con varianza adaptativa generó cinco zonas observación intermedias y la primera zona observación expuestas en la Tabla 3.10. El valor de varianza inicial de 0,050 fue reducido de forma automática hasta un valor de 0,010 para la primera zona observación. La segunda zona intermedia también sufre una fase de reducción en la varianza de forma que la varianza espacial para esta zona es de 0,025. Estas dos zonas se encuentran en las cercanías de la fuente y por tanto requieren menores valores de varianza. Siguiendo el proceso descrito en el diagrama de flujo 3.36, y haciendo uso de la primera zona observación, se generaron doce zonas fuente (ver Tabla 3.11). Además, existe una única zona frecuencial que ocupa todo el ancho de banda de trabajo de 9 GHz a 11 GHz. Gracias al mecanismo de búsqueda de varianza, los niveles de error son reducidos respecto a la situación de varianza fijada de forma heurística. Para conseguir una precisión mayor es necesario dividir el espacio en un número mayor de zonas tal como se aprecia si comparamos las Tablas 3.8-3.10 y 3.9-3.11. El algoritmo de división con varianza adaptativa permite, bien un menor umbral de error, o bien un número de zonas menor para un mismo nivel máximo de error.

Una vez dividido el espacio de entrada puede empezar el entrenamiento de las RBFNNs. En total son necesarias 73 RBFNNs para expresar totalmente las funciones de Green. Para lograr la precisión

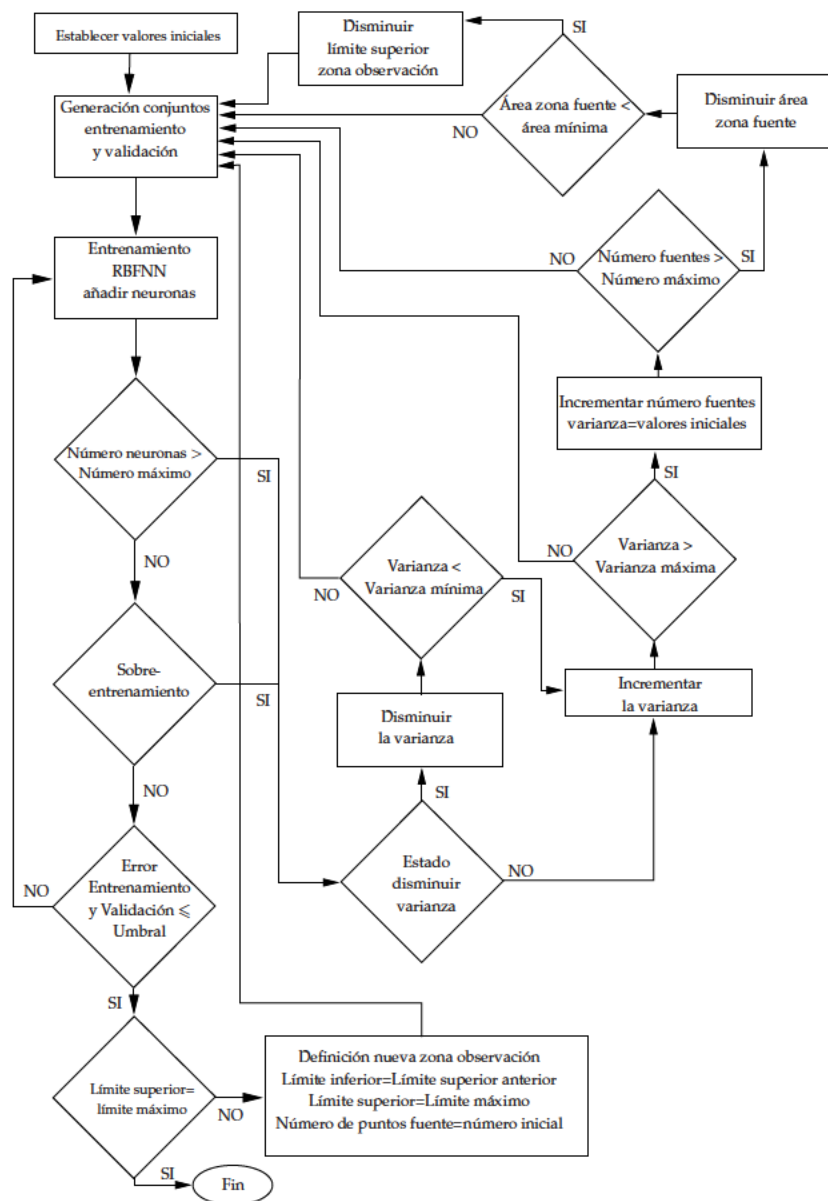


Figura 3.42: Diagrama de flujo del algoritmo de división en zonas de los puntos observación cuando se aplica el mecanismo de varianza adaptativa.

deseada fueron necesarios solo tres puntos en frecuencia situados en 9 GHz, 10 GHz y 11 GHz. La varianza de la coordenada frecuencial fue fijada a un valor $\sigma_f^2 = 10,0$. El valor de varianza de la coordenada frecuencial difiere mucho de las varianzas espaciales ya que el rango de variación de la coordenada frecuencial es a su vez diferente del rango de variación de las coordenadas espaciales. La asignación del valor medio del ancho de banda de trabajo a la varianza frecuencial ofrece buenos resultados en la aproximación en dicho ancho de banda. De nuevo, el uso de este sencillo esquema de división fue posible gracias a la ausencia de frecuencias de resonancia de la cavidad en el ancho de banda de trabajo.

Dependiendo de la cantidad de puntos fuente y observación, los conjuntos de entrenamiento de cada RBFNN poseen un tamaño que varía desde las 5000 muestras hasta más de 10000. Las RBFNNs

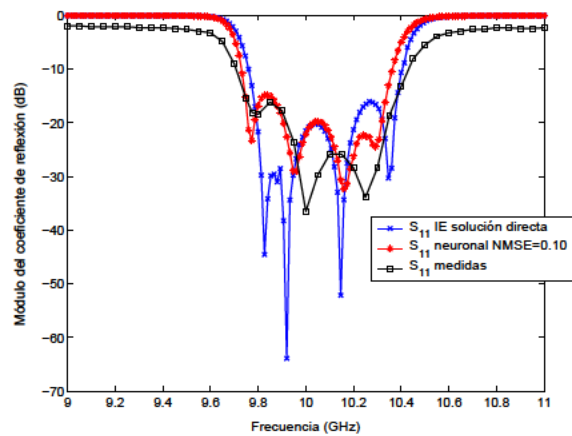
Tabla 3.10: Zonas observación generadas por el algoritmo de división con varianza adaptativa.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	Posición del punto fuente	0.0052	0.010
2ª zona	0.0052	0.0104	0.025
3ª zona	0.0104	0.0208	0.050
4ª zona	0.0208	0.0417	0.050
5ª zona	0.0417	0.0625	0.050
6ª zona	0.0625	0.1250	0.050
7ª zona	0.1250	1.0000	0.050

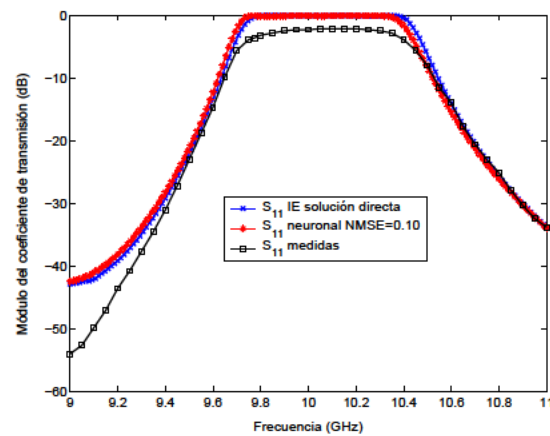
Tabla 3.11: Zonas fuente generadas por el algoritmo de división de fuentes con la primera zona observación definida con el algoritmo de varianza adaptativa.

Zona	Límite inferior	Límite superior
1ª zona	0.4000	0.5000
2ª zona	0.3000	0.4000
3ª zona	0.2000	0.3000
4ª zona	0.1438	0.2000
5ª zona	0.1200	0.1438
6ª zona	0.0102	0.1200
7ª zona	0.0844	0.0102
8ª zona	0.0669	0.0844
9ª zona	0.0491	0.0669
10ª zona	0.0313	0.0491
11ª zona	0.0135	0.0313
12ª zona	0.0000	0.0135

diseñadas fueron incorporadas a la formulación de la IE para analizar los filtros de líneas acopladas mostrados en las Figuras 3.37 y 3.38. En las Figuras 3.43 y 3.44 se muestra el módulo de los parámetros S_{11} y S_{21} de los filtros de cuarto orden impresos en la estructura multicapa de trabajo. Se aprecia una mejora sustancial en la respuesta simulada con el método neuronal. La banda de paso de la respuesta neuronal es prácticamente la misma que la obtenida con la simulación directa. Por el contrario, cuando se utilizaba las RBFNNs entrenadas con una varianza fija y común a todas las zonas, la banda de paso del método neuronal no se correspondía con la obtenida con la solución exacta. Asimismo, los niveles de atenuación conseguidos en la banda de paso para el parámetro S_{11} son similares a los alcanzados con la solución directa. Los polos de la respuesta correspondiente al método neuronal se sitúan cerca de los polos de la solución directa. Sin embargo, en ciertas zonas de la banda de paso el método neuronal no sigue con fidelidad las variaciones del parámetro S_{11} . Es necesario por tanto, incrementar la precisión del método para que la herramienta neuronal pueda ser empleada de forma efectiva en tareas de diseño. En este caso, la respuesta debe cumplir una serie de especificaciones; consecuentemente, si la simulación posee incorrecciones el diseño puede ser erróneo. En cuanto a los resultados neuronales para ambos filtros del parámetro S_{21} , éstos exhiben una gran semejanza con la

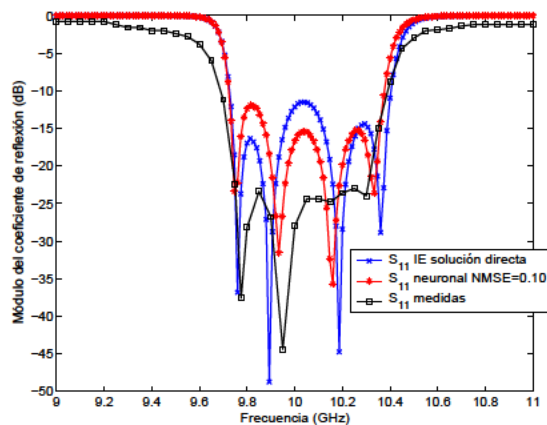


(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

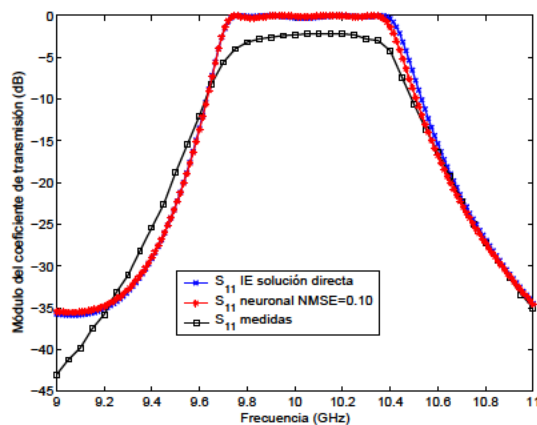


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

Figura 3.43: Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. Para validar la técnica se añaden valores correspondientes a medidas. La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.



(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

Figura 3.44: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. Para validar la técnica se añaden valores correspondientes a medidas. La respuesta se simula para 151 puntos de frecuencia.

respuesta que utiliza las funciones de Green evaluadas numéricamente. Para validar las simulaciones efectuadas se han añadido mediciones de ambos filtros como se aprecia en las Figuras 3.43 y 3.44.

Para ampliar el estudio del rendimiento del método neuronal se ha analizado un nuevo filtro de seis polos en la cavidad de trabajo (ver Figura 3.45). Este filtro formado por líneas acopladas presenta, al igual que los filtros de cuatro polos previos, acoplos críticos entre sus tiras metálicas. Los

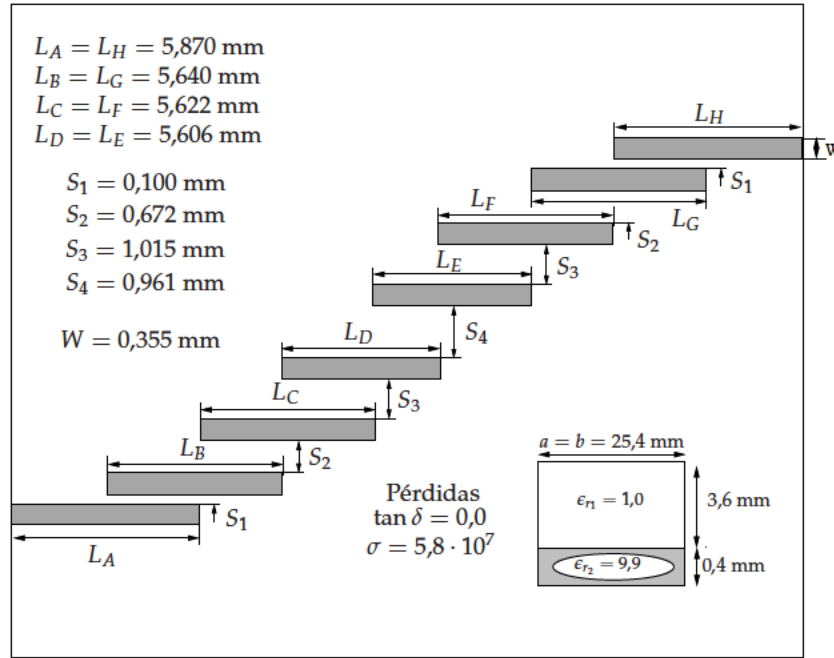
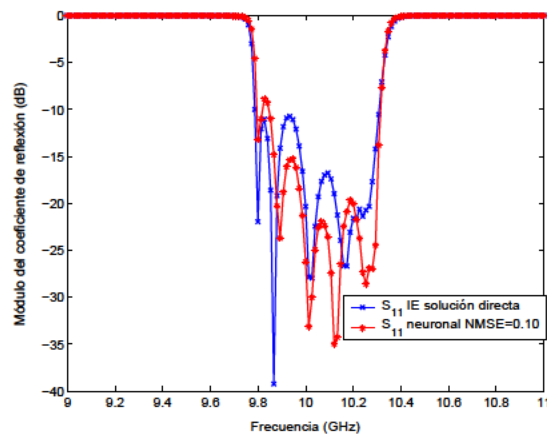


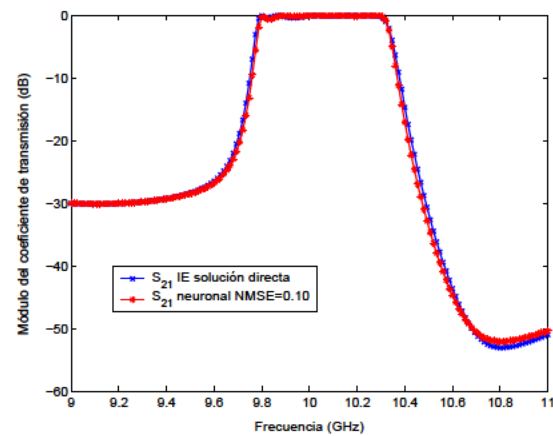
Figura 3.45: Filtro paso banda de seis polos. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).

parámetros S obtenidos con la solución exacta y con el método neuronal se muestran en las Figuras 3.46(a) y 3.46(b). Se puede observar que el método neuronal sigue con exactitud el decaimiento del parámetro S_{21} tanto en la parte inicial como en la parte final de la banda de paso. Por lo tanto, la respuesta neuronal no presenta un desplazamiento en frecuencia significativo respecto a la solución exacta, tal como ocurriría con el método neuronal que utilizaba una varianza fija para todas las zonas observación. No obstante, en la banda de paso, el método neuronal no sigue con exactitud las variaciones del parámetro de reflexión, Figura 3.46(a). El aumento del orden del filtro no implica una gran degradación de la precisión del método neuronal. Este hecho es debido a que las distancias entre tiras metálicas son similares entre los filtros de cuatro polos y el filtro de seis polos (la distancia mínima es de 0,100 mm para todos los filtros). Al igual que en los filtros de cuarto orden, el método neuronal presenta una respuesta correspondiente al parámetro S_{21} similar a la ofrecida por la solución exacta, Figura 3.46(b). El ejemplo presente reafirma la necesidad de incrementar la precisión de la aproximación de las funciones de Green.

Como ya se ha mencionado, el interés del método neuronal de análisis radica en la aceleración del cálculo de la respuesta de los circuitos impresos. La estrategia de división en zonas combinada con el procedimiento de varianza adaptativa permite una mayor precisión pero también provoca un aumento del número de zonas en las que se divide el espacio de entrada. Por consiguiente, el tiempo empleado en la selección de las distintas zonas y en el entrenamiento de las RBFNNs se incrementará en cierta medida. Si la ganancia temporal en el análisis circuital no es elevada o el tiempo de selección y entrenamiento es muy grande puede que el método neuronal se convierta en poco útil. Los resultados temporales resumidos en las Tablas 3.12 y 3.13 muestran en efecto que el tiempo necesario para diseñar las nuevas RBFNNs aumenta considerablemente. Mientras que el tiempo total de diseño en el caso de emplear una varianza común es de 14 horas, el periodo necesario para diseñar las nuevas



(a) Módulo de S_{11} del filtro paso banda de sexto orden mostrado en la Figura 3.45.



(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.45.

Figura 3.46: Respuesta del filtro de sexto orden de la Figura 3.45 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. La respuesta se simula para 151 puntos de frecuencia.

RBFNNs es de 81 horas. Sin embargo, la ganancia temporal conseguida con las nuevas RBFNNs más precisas es muy elevada y suficiente para considerar el tiempo de diseño insignificante a medida que nuevos filtros son analizados. Una vez todas las RBFNNs han sido entrenadas para una cavidad determinada y un ancho de banda de trabajo, no se requiere ningún entrenamiento adicional. En efecto, tanto el filtro de la Figura 3.37, como el filtro mostrado en 3.38, como el filtro de sexto orden de la Figura 3.45 son analizados con las mismas RBFNNs con un grado similar de precisión. Gracias al método neuronal la respuesta de cada uno de los anteriores filtros es conseguida en minutos en lugar de en horas. Con tan solo el análisis de los tres filtros estudiados el método neuronal resulta rentable. Si, además, se realizan diversos diseños circuitales en la misma estructura, el tiempo de diseño neuronal se vuelve despreciable. En cada diseño circuital es probable que sea necesario un proceso de optimización en el que la respuesta electromagnética deba ser simulada repetidas veces. En este caso, el método neuronal puede llegar a resultar muy atractivo para el diseño eficiente de circuitos impresos si se mejora en cierto grado su precisión. Para conseguir este objetivo se han desarrollado una serie de mejoras en la aproximación de las funciones de Green que se explican en la siguiente sección.

Tabla 3.12: Tiempo invertido en la selección de las zonas observación, fuente, frecuencia y tiempo de entrenamiento de las RBFNNs cuando se emplea el algoritmo con varianza fija.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	32 m.	3h. 20 m.	35 m.	9h. 39 m.	14 h. 6m.

Tabla 3.13: Tiempo invertido en la selección de las zonas observación, fuente, frecuencia y tiempo de entrenamiento de las RBFNNs cuando se emplea el algoritmo con varianza adaptativa.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	2h. 23 m.	10h. 33 m.	41 m.	68h. 05 m.	81 h. 42m.

Tabla 3.14: Comparación de los resultados temporales de la solución directa y el método neuronal con varianza adaptativa para el cálculo de los parámetros S en 151 puntos de frecuencia.

Filtro	Solución directa	Solución neuronal	Número veces más rápida
Cuatro polos Figura 3.37	112 h 27 m.	2445 segundos	165
Cuatro polos Figura 3.38	112 h 46 m.	2425 segundos	167
Seis polos Figura 3.45	124 h 49 m.	3170 segundos	141

3.9. Eliminación de la singularidad basada en la extracción de capas de imágenes espaciales. Zonas de observación circulares

En la sección precedente se han aplicado dos técnicas de forma combinada que permiten el desarrollo de una herramienta de análisis de circuitos encapsulados basada en RBFNNs. El primer mecanismo consiste en la extracción de la singularidad de la fuente mediante la multiplicación de las funciones de Green por la distancia (ρ) entre punto fuente y observación. La segunda estrategia se apoya en la división del espacio de entrada en zonas observación, fuente y frecuencia y la búsqueda de valores adecuados de varianza [Garcia et al., 2006b]. En esta Sección se van a aplicar dos nuevas estrategias que consiguen un aumento drástico de la precisión en la aproximación de las funciones de Green. A pesar de que la técnica empleada en [Garcia et al., 2006b] elimina de forma efectiva la singularidad en las zonas observación cercanas a la fuente, se observan todavía rápidas variaciones cerca de la mencionada fuente. Estas rápidas variaciones están producidas por la discontinuidad en las derivadas espaciales de las funciones de Green resultantes.

La primera estrategia de mejora en la aproximación de las funciones de Green radica en una nueva técnica de eliminación de la singularidad de la fuente. Gracias a la extracción de dos capas de imágenes en la formulación espacial mostrada en la Sección 2.3 se consigue una supresión total de la singularidad. Como se comprobará, las funciones de Green tratadas con esta técnica poseen un crecimiento lento incluso para aquellos puntos situados cerca de la fuente. Además, la singularidad previamente extraída es añadida de forma exacta a los valores de salida de las RBFNNs.

La segunda estrategia opera sobre las zonas en las que se divide el espacio de entrada. Cerca de la fuente los campos electromagnéticos presentan una simetría esférica debido al término dominante de la fuente [Melcon y Mosig, 2000]. En consecuencia, es más apropiado utilizar zonas observación circulares cerca de la fuente en lugar de zonas cuadradas como las dibujadas en la Figura 3.28.

Además, se realiza una división del espacio de entrada más profunda que la aplicada hasta aho-

ra. En la Sección 3.8 las zonas fuente eran calculadas únicamente con la zona de puntos observación situados más cerca de la fuente, es decir, con la denominada como primera zona observación. Estas zonas fuente eran asignadas al resto de zonas observación ya que la primera zona observación representa la situación más crítica. Gracias a la nueva técnica de extracción de la singularidad la primera zona observación no constituye el caso más difícil en la aproximación de las funciones de Green. Por tanto, es necesario un algoritmo de división más elaborado de modo que cada zona observación posea su propia colección de zonas fuente. La nueva división del espacio de entrada permite un incremento de la precisión y de la fiabilidad de la aproximación realizada.

En el Capítulo 2 se explicó el método empleado para evaluar numéricamente las funciones de Green. El plano transversal en el que se sitúan los puntos observación se dividía en dos zonas: la zona de máxima singularidad (MS) situada en torno a la fuente y la zona alejada de la fuente, denominada zona de débil singularidad (DS). En la zona MS se emplea el método híbrido de la Sección 2.3. Este método utiliza el mecanismo de suma de imágenes que se aplica a la parte cuasi-estática de la serie espectral que conforma las funciones de Green. El conjunto de imágenes espaciales se puede dividir en grupos de cuatro cargas o dipolos denominados conjuntos básicos de imágenes. Así, en la Sección 2.3 se definía un término k que indica el agrupamiento de un cierto número de conjuntos básicos de imágenes en la llamada capa de influencia. La relación entre los índices m y n de la serie original que forma las funciones de Green (ver ecuación (2.23)) y el término k se muestra en la Figura 2.5. De este modo, la capa de influencia $k = 1$ contiene un conjunto básico de imágenes que incluye el término fuente y las tres primeras imágenes. La siguiente capa de influencia correspondiente al término $k = 2$ contiene ocho conjuntos básicos. Estos dos términos, especialmente el primero, son responsables del comportamiento singular de las funciones de Green cerca de la fuente. Por lo tanto, la singularidad es eliminada de forma total si se extraen las dos primeras capas de influencia del sumatorio de imágenes espaciales. La suma de términos de imágenes espaciales sin singularidad, por consiguiente, comenzará a partir de $k = 3$. Una vez eludidos los dos primeros términos, la suma del resto de términos presenta un comportamiento suave, de forma que las RBFNNs podrán aproximar con facilidad las funciones de Green cerca de la fuente. Este método de extracción de la singularidad se aplica únicamente en la zona MS del espacio. En la zona DS las funciones de Green son multiplicadas por la distancia (ρ) entre fuente y observador, al igual que en la Sección previa. Aunque en esta zona no existe un comportamiento singular, la multiplicación por (ρ) permite que la evolución de las funciones de Green sea más "suave". De esta manera, la aproximación lejos de la fuente resulta más precisa si las funciones de Green son multiplicadas por la distancia entre fuente y punto observación. El umbral $U_x = U_y$ que divide las zonas MS y DS se fija de forma experimental según lo comentado en la Sección 2.4.

Con la nueva estrategia de eliminación de la singularidad, los valores finales utilizados en la IE son obtenidos mediante la adición, a los valores de salida de las RBFNNs, de los términos correspondientes a dos capas de influencia. Estos términos representan de forma exacta la singularidad, así la precisión en la expresión de las funciones de Green aumenta de forma drástica. La operación de recuperación no representa un coste computacional elevado, debido a que los términos que se deben sumar están compuestos por una suma analítica finita y reducida de imágenes espaciales. Además, la técnica de extracción de la singularidad empleada en las Secciones 3.7 y 3.8 (multiplicación por la distancia (ρ)) también tiene cierta carga computacional extra, ya que se debe calcular la distancia

entre fuente y punto observación para cada valor de salida. Para mostrar la habilidad de la nueva técnica de extracción en la generación de funciones de lento crecimiento en torno a la fuente, la función de Green del potencial eléctrico escalar (G_V) se ha calculado en dos situaciones diferentes. La cavidad de trabajo es la representada en las Figuras 3.37 y 3.38 y la frecuencia es 11 GHz. La primera situación corresponde a un punto fuente A ubicado cerca del centro de la cavidad de la Figura 3.47, por tanto lejos de las paredes metálicas. Una línea recta de puntos observación situada sobre la fuente es seleccionada con el objetivo de mostrar el comportamiento de G_V en esta situación crítica. En la Figura 3.48(a) se aprecia el suave comportamiento de la función de Green cuando se extraen dos capas de influencia en contraste con la variación rápida provocado con la técnica previa. En particular, con la anterior técnica se observa la naturaleza discontinua de las derivadas espaciales en el punto de cruce de la fuente. Es razonable deducir que la aproximación de las funciones resultantes de la aplicación de la nueva técnica proporcionará menores niveles de error. En la segunda situación estudiada, el punto fuente se coloca cerca de las paredes metálicas, en la posición B de la Figura 3.47. Una vez más, la nueva técnica produce una función de lenta variación a lo largo de la línea de puntos observación tal y como se ve en la Figura 3.48(b). Por el contrario, la función obtenida con la técnica de multiplicación por (ρ) produce variaciones rápidas en el punto fuente. La gráfica de la Figura 3.48(b) muestra que la nueva estrategia de eliminación de la singularidad mejorará la precisión también en aquellas zonas fuente situadas cerca de las paredes. Para mostrar de forma más completa la evolución de las funciones de Green en toda la zona MS cuando se extraen dos capas de influencia, podemos recurrir a dos simulaciones similares a las expuestas en las Figuras 3.14 y 3.15. Así, en las Figuras 3.49 y 3.50 se exhibe el valor de G_V sin singularidad para una serie de puntos de observación situados de forma regular en torno a los dos puntos fuente de la Figura 3.47. La función resultante muestra una variación con ausencia total de brusquedad tanto si el punto fuente se sitúa lejos de las paredes laterales (Figura 3.49) como si se ubica cerca de las paredes metálicas (Figura 3.50).

Es un hecho conocido que cerca del punto fuente el comportamiento de los campos electromagnéticos tiende a ser esférico, debido a la simetría impuesta por la propia fuente puntual. Por lo tanto, para expresar de forma más correcta las variaciones de las funciones de Green, es conveniente dividir el espacio en zonas observación circulares cerca de la fuente, como se observa en la Figura 3.51. Por el contrario, lejos de la fuente las paredes metálicas imponen una simetría de tipo rectangular en las funciones de Green. En consecuencia, la parte del espacio de entrada correspondiente a aquellos puntos observación localizados lejos de la fuente se agruparán en zonas cuadradas, como en el algoritmo desarrollado en la Sección 3.8. Así, se emplearán zonas circulares en la zona MS, cercana a la fuente, mientras que en la zona DS se utilizarán las zonas cuadradas.

Las paredes metálicas influyen en la simetría esférica cuando el punto fuente está cerca de las paredes. A pesar de dicho efecto, la división circular se mantiene para estos puntos fuente ya que la influencia de la fuente es más fuerte que la ejercida por las paredes metálicas. Se debe tener en cuenta que en el caso de que la fuente se sitúe cerca de una pared metálica la zona circular debe contener únicamente puntos observación dentro de la cavidad. Los puntos observación situados fuera de la cavidad no son considerados y así las zonas circulares se transforman en zonas semi-circulares como se aprecia en la Figura 3.51. El conjunto de zonas observación se completa con una zona de transición entre las zonas circulares y las zonas cuadradas que forman la región DS del espacio. El límite exterior de la zona de transición coincide con el umbral que separa la región MS de la región DS como se

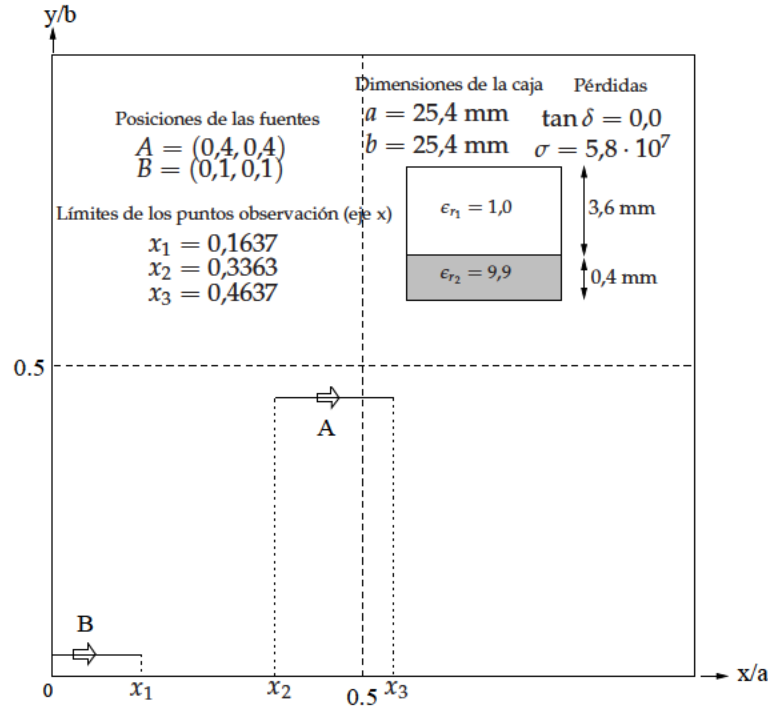
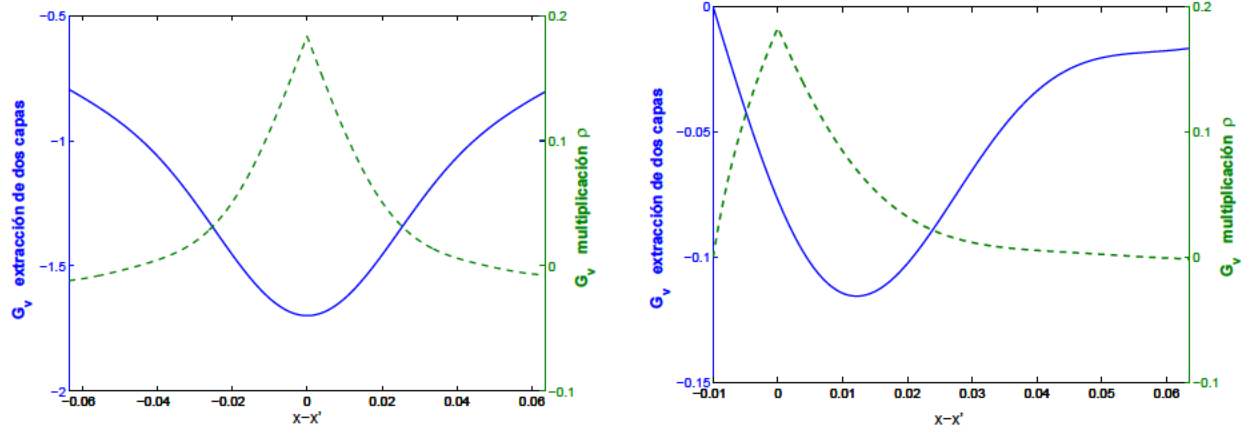


Figura 3.47: Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado cerca del centro del plano transversal. Misma estructura de la Figura 2.7 de la Sección 2.3 y utilizada en los filtros 3.37, 3.38 y 3.45.



(a) Función de Green del potencial eléctrico escalar G_V cuando la singularidad es extraída. La fuente se sitúa en la posición A de la Figura 3.47. La línea sólida (valores en el eje izquierdo) corresponden a la nueva técnica de extracción (expresado como G_V). La línea de guiones (valores en el eje derecho) corresponden a la técnica de la Sección 3.8 (expresado como ρG_V).

(b) Función de Green del potencial eléctrico escalar G_V cuando la singularidad es extraída. La fuente se sitúa en la posición B de la Figura 3.47. La línea sólida (valores en el eje izquierdo) corresponden a la nueva técnica de extracción (expresado como G_V). La línea de guiones (valores en el eje derecho) corresponden a la técnica de la Sección 3.8 (expresado como ρG_V).

Figura 3.48: Comparativa entre las dos técnicas de eliminación de la singularidad desarrolladas.

observa en la Figura 3.51. Dependiendo de la adscripción de la zona observación a la región MS o a

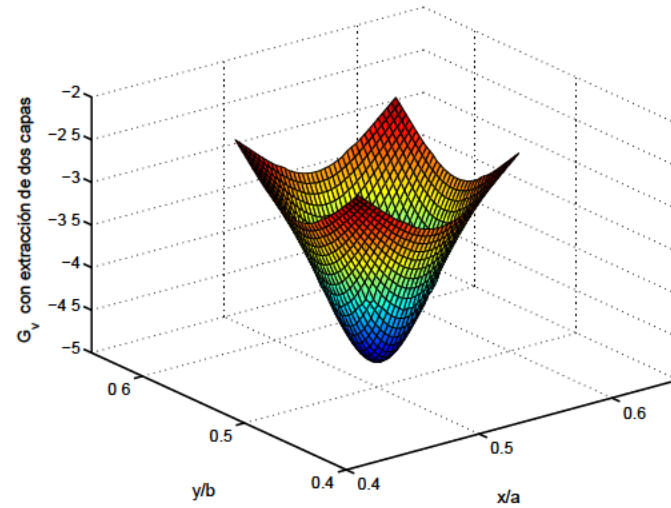


Figura 3.49: Efecto de la extracción de dos capas de influencia en la función de Green del potencial eléctrico escalar. Punto fuente situado cerca del centro de la cavidad representada en la Figura 3.47.

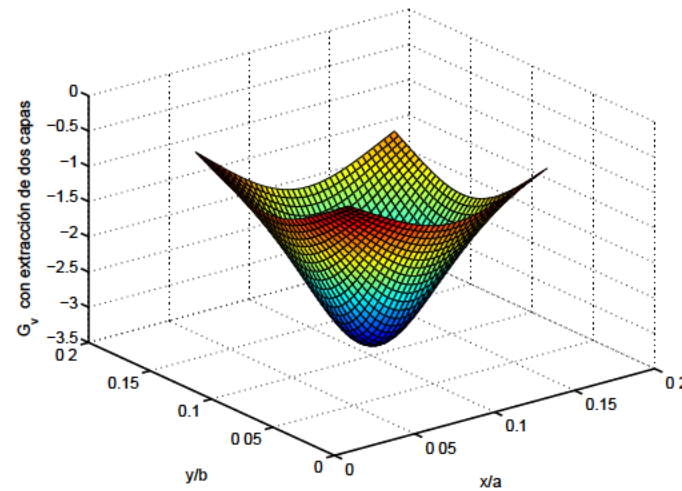


Figura 3.50: Efecto de la extracción de dos capas de influencia en la función de Green del potencial eléctrico escalar. Punto fuente situado cerca de las paredes de la cavidad representada en la Figura 3.47.

la región DS las funciones de Green sufrirán una transformación diferente:

- Zonas circulares: estas zonas pertenecen a la zona MS. La singularidad debe ser eliminada por el método de la extracción de dos capas de influencia.
- Zona de transición: se encuentra en el interior de la región MS. La influencia de la singularidad comienza a ser débil. Por tanto, las funciones de Green en esta zona pueden ser multiplicadas por (ρ) si este procedimiento logra un nivel de error de aproximación menor que en el caso de

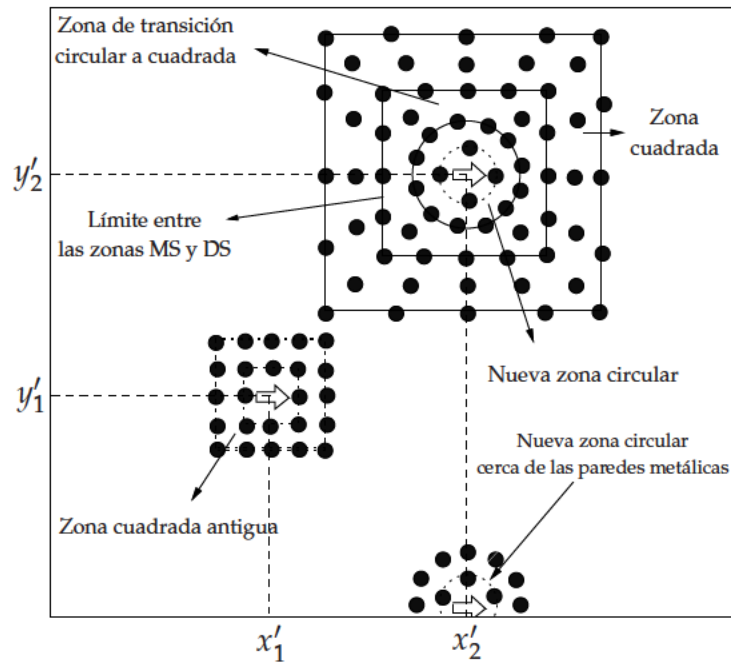


Figura 3.51: Ejemplos de zonas circulares y cuadradas. Los puntos negros simbolizan puntos observación.

emplear la extracción de dos capas de influencia.

- Zonas cuadradas: se emplean en la región DS. Las funciones de Green se multiplican por la distancia (ρ) ya que consigue una mayor suavidad con el consiguiente aumento de la precisión.

El algoritmo de división de zonas observación descrito en la Sección 3.8 se aplica de forma análoga a las zonas observación de la Figura 3.51. La única diferencia estriba en la aplicación de una fase de incremento de puntos de observación en las zonas circulares. Este paso tiene por objetivo el aumento de la densidad de puntos de forma combinada en la zona observación y fuente. En el algoritmo de la Figura 3.42 la cantidad de puntos de las zonas intermedias se mantenía fijo, de forma que la densidad solo era ampliada mediante la reducción del área de la zona observación bajo estudio. Ahora, las zonas observación circulares poseen mayor libertad para distribuir los puntos observación. El aumento de puntos se puede llevar a cabo bien con el incremento del número de anillos bien con el aumento del número de puntos por anillo. Como se explicó en la Sección 3.8 el proceso de determinación de las zonas observación define a su vez los límites de la primera zona fuente. El resto de zonas fuente se definían haciendo uso de la primera zona observación. Gracias a la nueva técnica de eliminación de la singularidad, la aproximación de las funciones en los puntos observación cercanos a la fuente es sencilla. La división de los puntos fuente llevada a cabo con la zona observación más cercana a la fuente no resulta apropiada para el resto de zonas observación. Consecuentemente, el proceso iterativo descrito en el diagrama de flujo de la Figura 3.36 para la división de las zonas fuente es aplicado a todas las zonas observación. La obligatoria ampliación del proceso de división de las fuente origina beneficios adicionales ya que tiene en cuenta el comportamiento de las funciones de Green, tanto lejos como cerca de las paredes para todas las zonas observación.

3.9.1. Versión mejorada del algoritmo OLS

Lógicamente, el proceso global de división del espacio de entrada consumirá más tiempo que en el caso de los métodos de la Sección 3.8. Podemos disminuir el tiempo de selección de zonas y entrenamiento si reducimos el tiempo de entrenamiento de la red neuronal sin perder exactitud en la aproximación. Existen diversas versiones del algoritmo de entrenamiento OLS que permiten importantes reducciones en el tiempo empleado en la selección de los centros [Chng et al., 1995]. La versión más rápida de este algoritmo se denomina "Fast orthogonal least squares" (FOLS) y se encuentra publicada en [Chen y Wigger, 1995]. La mayor parte del coste computacional del algoritmo OLS se debe a la realización de productos escalares entre vectores, tales como el producto entre regresores $(\mathbf{m}_k^i)^T \mathbf{m}_k^i$, o como el producto entre un regresor y el vector de valores de salida $((\mathbf{m}_k^i)^T \mathbf{y})^2$. Así ocurre en el algoritmo con simplificaciones de la Sección 3.5 como se observa en las ecuaciones (3.57), (3.58), (3.59) y (3.60). Asimismo, en el caso de que el espacio de salida sea multidimensional, se deberán llevar a cabo una serie de productos entre regresores además de productos entre los vectores columna de la matriz de datos de salida y los regresores, como se ve en la ecuación (3.61).

En la versión rápida desarrollada en [Chen y Wigger, 1995], en lugar de actualizar los regresores y calcular en la siguiente iteración los productos escalares, se procede a la actualización directa de dichos productos escalares. Las gráficas expuestas en [Chen y Wigger, 1995] demuestran que para un determinado número de datos de entrenamiento existe un número de neuronas a partir del cual el algoritmo FOLS es más rápido que el algoritmo clásico OLS. Recuperando los datos de [Chen y Wigger, 1995] observamos que si el conjunto de entrenamiento posee 500 datos con un número de candidatos igual, el algoritmo FOLS requiere un número significativamente menor de operaciones para seleccionar 100 o más regresores. En nuestro problema, contamos con conjuntos de entrenamiento formados por varios miles de muestras. Por otra parte, el número de neuronas debe ser limitado a pocos cientos para acelerar lo máximo posible el análisis circuital mediante la IE. El algoritmo FOLS consigue una gran reducción de la carga computacional a partir de un número de neuronas demasiado elevado cuando el conjunto de entrenamiento está formado por miles de muestras. Con el objetivo de reducir el número de operaciones respecto al algoritmo simplificado de la Sección 3.5, a partir de una cantidad limitada de neuronas, se ha desarrollado una nueva versión del algoritmo OLS. Esta versión guarda solo ciertos productos escalares en lugar de almacenar el conjunto total de productos como ocurre en el algoritmo FOLS. Esta limitación en el almacenamiento y actualización de los productos escalares permite que la nueva versión sea más rápida que el algoritmo FOLS para un número reducido de neuronas, en comparación con el tamaño del conjunto de entrenamiento.

Concretamente la versión mejorada del algoritmo OLS almacena los productos escalares de los regresores con los vectores columna de la matriz de valores de salida. Si el espacio de salida es unidimensional, la matriz de salida se transforma en un vector columna. En general el espacio de salida será de dimensión q y el número de candidatos \mathbf{M}_s será igual al número de datos de entrenamiento N . Así, los productos escalares $(\mathbf{m}_k^i)^T \mathbf{y}_\eta$ de la ecuación (3.61) para $1 \leq i \leq N$ y $1 \leq \eta \leq q$ se almacenan en la matriz de dimensión $N \times q$:

$$\mathbf{M}_y = \mathbf{M} \mathbf{y} \quad (3.63)$$

De esta forma cuando se calcule la reducción del error mediante la ecuación (3.61), en lugar de calcular el producto escalar entre el regresor i -ésimo y el vector columna j -ésimo de la matriz \mathbf{y} , simplemente se recuperará el valor de la matriz \mathbf{M}_y de la posición (i, η) . Ahora bien, siguiendo el proceso de las ecuaciones (3.57), (3.58) y (3.59) es necesario transformar aquellos regresores no seleccionados al final de cada iteración. En consecuencia, es obligatorio alterar los valores de la matriz \mathbf{M}_y para que los productos escalares almacenados reflejen la transformación de los mencionados regresores. Así, una vez seleccionado el regresor k -ésimo, \mathbf{m}_k , se debe realizar la siguiente operación:

$$\mathbf{M}_y(i, \eta)_{k+1} = (\mathbf{m}_{k+1}^i)^T \mathbf{y}_\eta = \left[\mathbf{m}_k^i - \frac{\mathbf{m}_k^T \mathbf{m}_k^i}{\mathbf{m}_k^T \mathbf{m}_k} \mathbf{m}_k \right]^T \mathbf{y}_\eta = \mathbf{M}_y(i, \eta)_k - \left[\frac{\mathbf{m}_k^T \mathbf{m}_k^i}{\mathbf{m}_k^T \mathbf{m}_k} \mathbf{m}_k \right]^T \mathbf{y}_\eta$$

para $1 \leq i \leq M_s = N$ $i \neq i_1 \neq i_2 \cdots \neq i_k$ y $1 \leq \eta \leq q$ (3.64a)

El vector $\frac{\mathbf{m}_k^T \mathbf{m}_k^i}{\mathbf{m}_k^T \mathbf{m}_k} \mathbf{m}_k$ de la ecuación (3.64a) también debe calcularse para realizar la actualización de los regresores no seleccionados. Por tanto, no representa ninguna carga computacional adicional respecto al algoritmo OLS clásico. Gracias a este hecho, el algoritmo mejorado consigue una gran reducción en el número de operaciones para cualquier número de neuronas M . La única serie de operaciones adicionales significativas son las multiplicaciones realizadas para evaluar el producto escalar entre el vector anterior y el vector columna \mathbf{y}_η .

Siguiendo la metodología aplicada en el trabajo [Chen y Wigger, 1995], el rendimiento de los algoritmos OLS, FOLS y la nueva versión se compara mediante el número de multiplicaciones realizadas en cada método. El algoritmo FOLS en [Chen y Wigger, 1995] está desarrollado solo para espacios de salida unidimensionales. Si consideramos un espacio de salida multidimensional el número de multiplicaciones necesarias para elegir M regresores es:

$$\begin{aligned} \text{Numero multiplicaciones FOLS} &= \frac{N(M_s + q)(M_s + q)}{2} - M(q + 1)q \\ &+ \sum_{p=1}^M [(M_s + p - 1)(3q + 1) + (M_s + q - p)(M_s + q - p + 2)] \end{aligned}$$

(3.65a)

donde N es el número de muestras del conjunto de entrenamiento, M_s es el número del conjunto inicial de regresores, M es el número de regresores seleccionados y q es la dimensión del espacio de salida. El número de multiplicaciones del algoritmo OLS clásico se ha calculado aplicando las simplificaciones de la Sección 3.5:

$$\text{Numero multiplicaciones OLS} = Nq - M(2N + 1) + \sum_{p=1}^M [(2N + 1 + (N + 1)(q + 1)) (M_s - p + 1)]$$

(3.66)

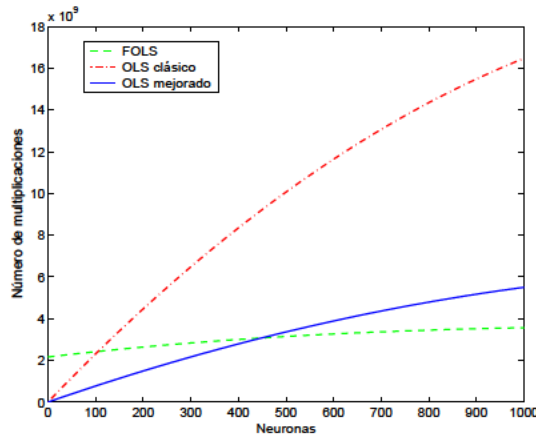
Finalmente, el número de multiplicaciones requeridas por la versión mejorada del algoritmo OLS es:

$$\text{Numero multiplicaciones OLS mejorado} = Nq - M(2N + q + 1) + \sum_{p=1}^M [(M_s - p + 1)(3N + 2q + 2)] \quad (3.67)$$

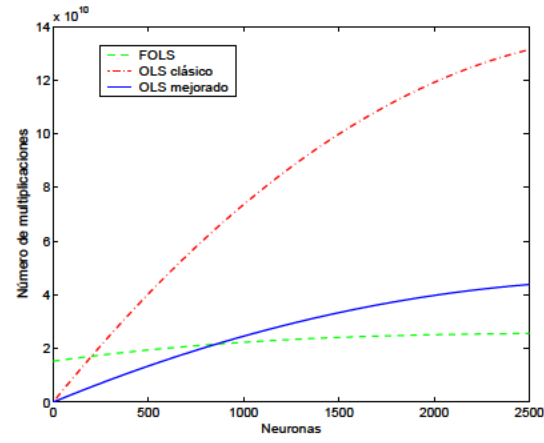
En la serie de Figuras 3.52 se observa el número de multiplicaciones para varios valores de $N = M_s$. En los ejemplos prácticos de las Secciones 3.7 y 3.8 el tamaño máximo de cada RBFNN se fijó en 700 neuronas. De esta forma, en el caso de la caja mostrada en la Figura 3.7 las RBFNNs definidas con el algoritmo de varianza fija poseían un tamaño medio de 511 neuronas. Como se ve en la Figura 3.52(a) para un número relativamente pequeño de datos de entrenamiento, el algoritmo mejorado consigue una eficiencia mayor que el método FOLS dentro del rango de neuronas de interés (entre 400 y 700 neuronas). Conforme aumenta el tamaño del conjunto de entrenamiento el algoritmo mejorado aumenta su eficacia respecto a los otros dos algoritmos. El tamaño máximo de entrenamiento utilizado en los algoritmos de división de zonas observación y fuente se sitúa en torno a las 3000 muestras. Como se aprecia en la Figura 3.52(b) si el conjunto de entrenamiento está formado por 3120 datos, el algoritmo FOLS no supera al algoritmo mejorado sino a partir de un número de neuronas de 900 unidades. Ésta es una cantidad muy superior al límite de neuronas impuesto a todas las redes neuronales. Por consiguiente, en los procedimientos de división de zonas observación y fuente, el algoritmo mejorado será siempre mucho más veloz que el método FOLS. Durante el entrenamiento de las RBFNNs es indispensable muestrear de forma suficiente cada una de las zonas. Si la coordenada frecuencia se muestrea con tres puntos es habitual encontrar conjuntos de entrenamiento con tamaños que oscilan entre las 7000 y las 9000 muestras. En las Figuras 3.52(c) y 3.52(d) se observa que dentro de los márgenes en los que se establece el tamaño neuronal de las RBFNNs, el algoritmo mejorado es mucho más rápido que el algoritmo FOLS. Por lo tanto, la versión del algoritmo OLS desarrollada habilitará una completa división del espacio de entrada sin que el tiempo total de división y entrenamiento convierta en poco práctico el método neuronal

3.9.2. Test del rendimiento de las nuevas técnicas en la aproximación de las funciones de Green

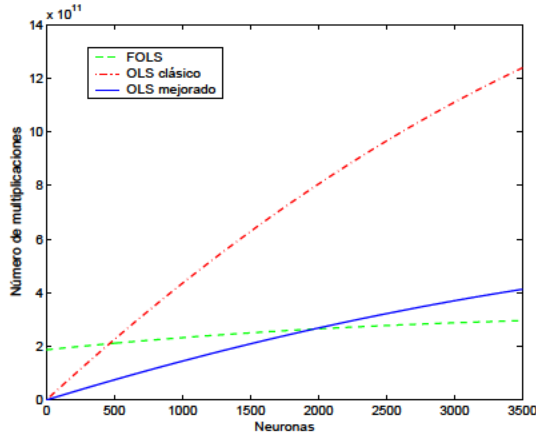
Las nuevas técnicas deben producir un gran aumento en la precisión de la aproximación de las funciones de Green. Para comprobar la capacidad de la nueva técnica de extracción de la singularidad y de las nuevas zonas circulares se ha llevado a cabo un test en dos cajas diferentes. En cada cavidad se han utilizado diferentes combinaciones de zonas observación y fuente para generar los conjuntos de entrenamiento y validación. Dos zonas fuente han sido elegidas dependiendo de su proximidad a las paredes metálicas. La primera zona fuente se encuentra alejada de las paredes, mientras que la segunda zona fuente tiene forma de "L" y se sitúa cerca de las paredes, como se observa en la Figura 3.53. Las nuevas estrategias se van a aplicar en la región MS, por tanto cerca de la fuente. Para demostrar las ventajas del novedoso método de cancelación de la singularidad, las zonas observación utilizadas en este test corresponden a la zona más cercana a la fuente. Con el objeto de ampliar el estudio sobre el rendimiento, han sido empleadas dos zonas observación de diferente tamaño. Además,



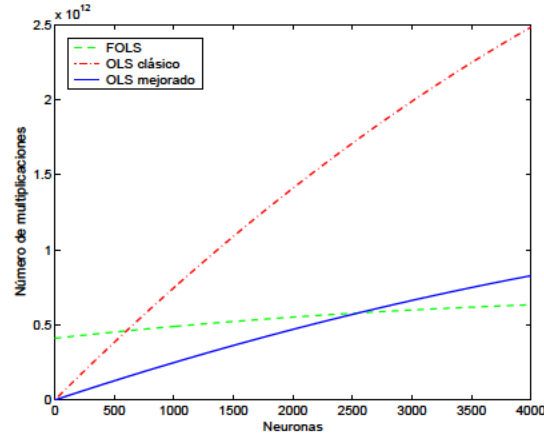
(a) Número de datos de entrenamiento igual a 1625.



(b) Número de datos de entrenamiento igual a 3120.



(c) Número de datos de entrenamiento igual a 7203.



(d) Número de datos de entrenamiento igual a 9360.

Figura 3.52: Número de multiplicaciones para los algoritmos FOLS, OLS clásico y OLS mejorado para diferentes tamaños del conjunto de entrenamiento N .

cada zona puede poseer forma cuadrada o circular. Los lados de las zonas cuadradas han sido ajustados para que el área de dicha zona sea aproximadamente igual al área de la correspondiente zona circular. La zona de menor tamaño ocupa un área normalizada igual a $7,9 \cdot 10^{-6}$, mientras que la zona mayor tiene un área normalizada igual a $7,9 \cdot 10^{-4}$.

Las zonas fuente y observación son llenadas gradualmente con puntos fuente y observación dando lugar a distintos conjuntos de entrenamiento y validación. En estos conjuntos aparecen desde zonas observación y fuente muestreadas de forma densa hasta otras escasamente muestreadas. Una plétora de casos intermedios tanto para la zona observación como para la zona fuente es cubierta. En la aproximación de cada combinación de zona observación, zona fuente y técnica de aproximación se utiliza una serie de varianzas que varían desde $\sigma^2 = 0,00625$ hasta $\sigma^2 = 0,1$. Este rango de varianzas es suficientemente amplio para garantizar una aproximación adecuada tanto para la zona observación pequeña como para la grande, así como para las distintas estrategias de aproximación. En cada caso, los resultados mostrados a continuación corresponden a los que ofrecen el menor error en término medio dentro del rango de varianzas explorado. De este modo, la varianza no influye en la comparación entre las diferentes técnicas empleadas. La precisión se ha medido en todas los casos

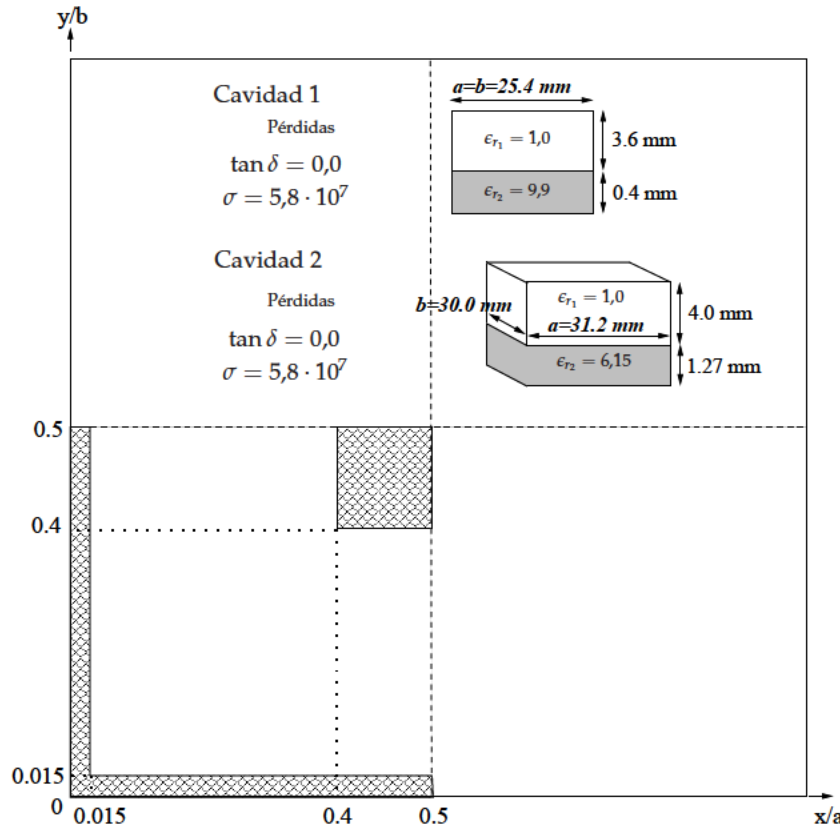


Figura 3.53: Zonas fuente y cavidades metálicas empleadas para testear las técnicas de aproximación de las funciones de Green.

mediante el error cuadrático medio. Este tipo de error permite una visión más clara de la evolución de la precisión en las técnicas estudiadas. En este tipo de test, para una misma técnica de aproximación, el error NMSE en las zonas fuente y observación muestreadas pobremente es mucho más elevado que en aquellos casos de mayor densidad de muestreo. Así, incluso empleando una escala logarítmica la comparación no resultaba clara.

En las Figuras 3.54 y 3.61 se muestran los resultados de los tests realizados. En todas ellas la línea dibujada con puntos y guiones corresponde a los resultados obtenidos al aplicar la técnica de eliminación de la singularidad mediante la extracción de dos capas en combinación con zonas observación circulares, la línea continua corresponde a la técnica de extracción de dos capas en combinación con zonas cuadradas, la línea con guiones corresponde a la técnica de eliminación de la singularidad mediante la multiplicación por (ρ) en combinación con zonas circulares, finalmente la línea con puntos corresponde a la técnica de eliminación de la singularidad mediante la multiplicación por (ρ) en combinación con zonas cuadradas.

En la Figura 3.54 se exhibe el error cuadrático medio correspondiente a la zona observación de mayor tamaño y a la zona fuente alejada de las paredes para las dos estructuras de trabajo. Se aprecia el gran aumento en la precisión de la aproximación cuando se emplea la técnica de extracción de dos capas de influencia. Las zonas circulares permiten una ligera mejora tanto si se emplea la nueva estrategia de eliminación de la singularidad como en el caso de utilizar la multiplicación por

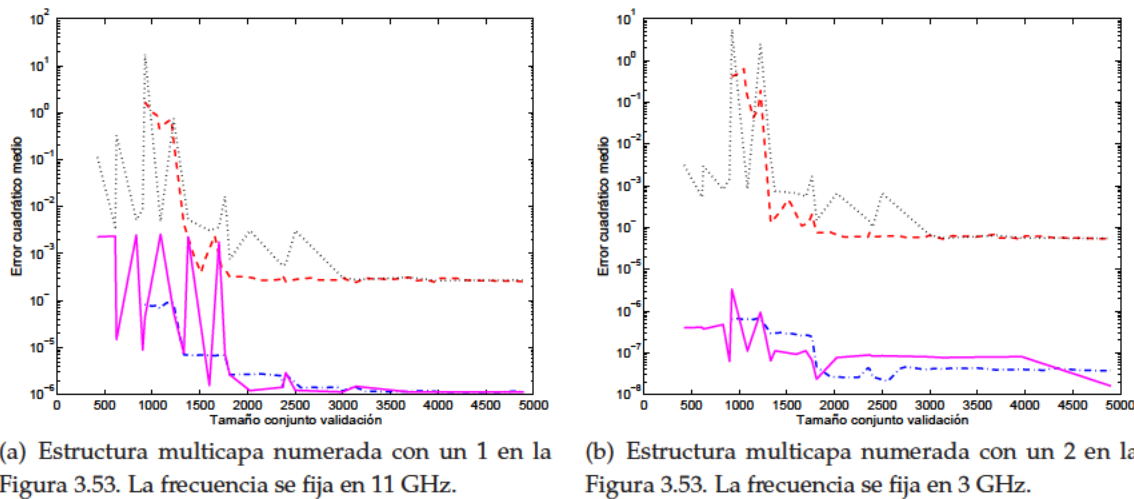


Figura 3.54: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$.

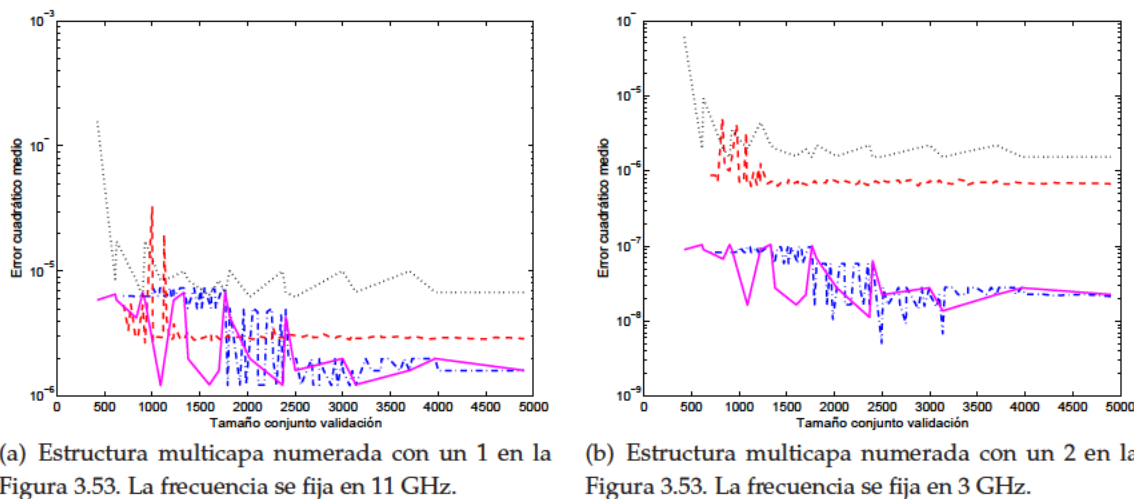


Figura 3.55: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$.

la distancia (ρ). Dicha mejora, debida a las zonas circulares, es más notable en la segunda cavidad. Además, tanto en la primera cavidad como en la segunda la mejora producida por las nuevas zonas se produce con tamaños de entrenamiento intermedios.

En el caso de reducir el tamaño de la zona observación se obtienen resultados similares tal como se observa en la Figura 3.55. Al igual que en el caso anterior, la zona fuente se encuentra lejos de las paredes. En general, existe un descenso del error conforme aumenta el tamaño de entrenamiento. Sin embargo, a partir de cierto punto el aumento del número de muestras del conjunto de entrenamiento no produce un descenso claro del error. Esto es debido a que los valores mostrados corresponden a aquel valor de varianza que produce el menor error en término medio. Cuando las zonas observación

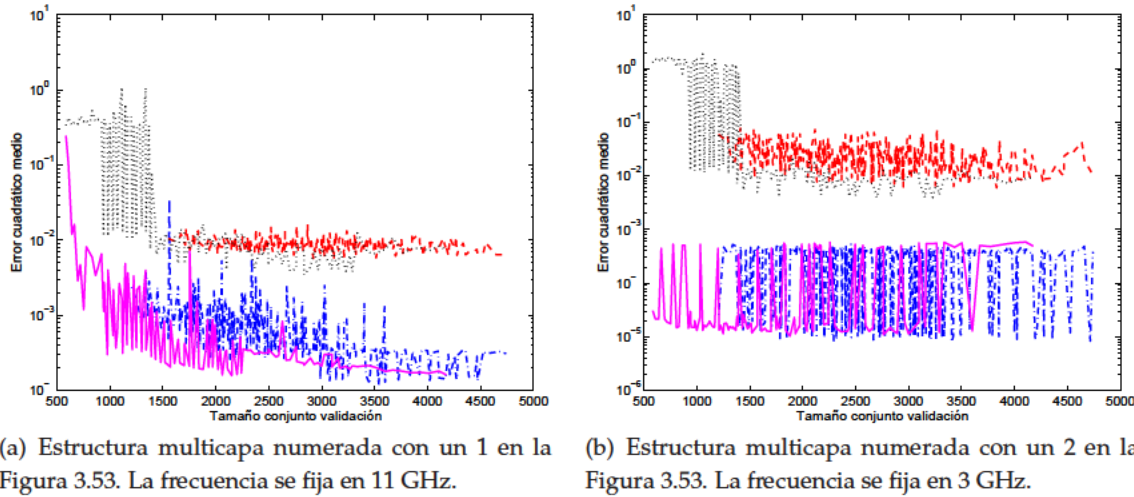
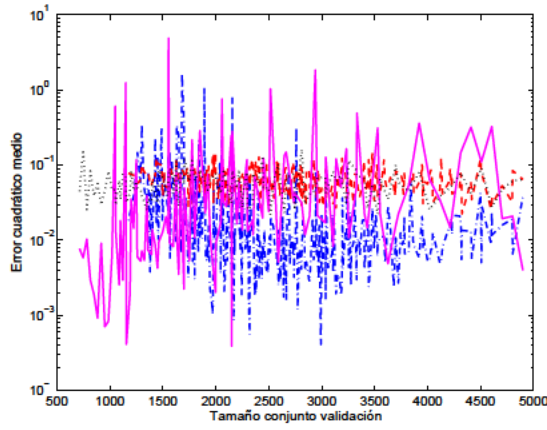


Figura 3.56: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$.

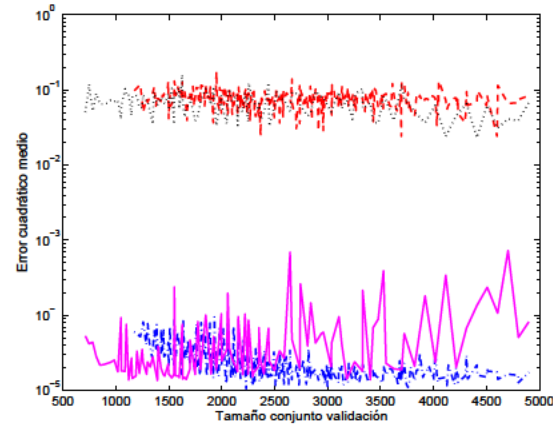
y fuente son muestreadas de modo más denso es más apropiado utilizar varianzas más pequeñas. Estas varianzas producirán mayores errores para las combinaciones de zonas observación y fuente con menor número de muestras, de modo que el rendimiento general se verá degradado. El rango de varianzas utilizado no cubre valores muy extremos ya que éstos no son adecuados para la ejecución del algoritmo de división con varianza adaptativa. Un valor demasiado grande de varianza no alcanzará en ninguna situación el valor de error fijado, aumentando el tiempo total de selección de forma innecesaria. Un valor muy reducido necesitará zonas con gran densidad de muestras. Estas zonas producirán conjuntos de entrenamiento de gran tamaño de imposible empleo debido a limitaciones en la memoria del ordenador.

Respecto a la comparación entre las técnicas estudiadas, de nuevo el nuevo procedimiento de eliminación de la singularidad ofrece la mayor precisión. La combinación de esta técnica con las zonas de tipo circular permite una pequeña mejora en la aproximación de las funciones de Green respecto a la utilización de las zonas cuadradas. La principal mejora en la generación de los conjuntos de entrenamiento viene dada por la extracción de dos capas de influencia. Las zonas circulares consiguen un descenso del error significativo en el caso de emplear la técnica de multiplicación por (ρ) . La incidencia de este tipo de zonas es mayor cuando se emplea una técnica de eliminación de la singularidad menos efectiva.

En la Figura 3.56 se exponen los errores cuando la zona fuente se localiza cerca de las paredes y la zona observación es de mayor tamaño. Tanto en la cavidad 1 como en la cavidad 2 la aproximación se torna en más complicada, ya que el error cuadrático medio es mayor que en la Figura 3.54. Como en el caso anterior, la eliminación de las dos primeras capas de influencia produce el mayor descenso en los niveles de error. La combinación con zonas circulares produce un pequeño incremento en la precisión general. En el caso de emplear la técnica de multiplicación por (ρ) , las zonas circulares no consiguen una clara ventaja frente a las zonas cuadradas. La influencia de las paredes provoca cierta pérdida en la simetría esférica de las funciones de Green. Sin embargo, las zonas circulares producen errores



(a) Estructura multicapa numerada con un 1 en la Figura 3.53. La frecuencia se fija en 11 GHz.



(b) Estructura multicapa numerada con un 2 en la Figura 3.53. La frecuencia se fija en 3 GHz.

Figura 3.57: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$.

de nivel similar o incluso menor a los alcanzados con las zonas cuadradas en ambas estrategias de eliminación de la singularidad.

Finalmente, se muestran los resultados correspondientes a la zona observación de menor área y la zona fuente cercana a las paredes metálicas para ambas cavidades. En este caso, se alcanzan las mismas conclusiones que en las situaciones anteriores. Los nuevos procedimientos de aproximación alcanzan una precisión mayor que las técnica de multiplicación por (ρ) y las zonas cuadradas para todos los tamaños de entrenamiento estudiados.

Como hemos visto en las Figuras 3.54, 3.55, 3.56 y 3.57, la nueva técnica de eliminación de la singularidad alcanza una gran reducción en los niveles de error de aproximación de las funciones de Green sin singularidad. La gran ventaja de la nueva técnica de eliminación de la singularidad radica en la recuperación exacta de la singularidad. De este modo, el error cometido por la RBFNN en la aproximación de las funciones de Green sin singularidad es el mismo que se comete cuando se recupera dicha singularidad. La técnica de multiplicación por (ρ) , por el contrario, amplifica el error cometido por la RBFNN al dividir por la mencionada distancia (ρ) para recuperar la singularidad. En las siguientes figuras se aprecia claramente este hecho. En las Figuras 3.58 y 3.59 se observa que el error de las funciones de Green una vez recuperada la singularidad con la antigua técnica es hasta ocho y nueve órdenes de magnitud mayor que el error cuando se emplea la técnica de extracción de dos capas de influencia. Además, se aprecia que los niveles de error en este último caso son iguales a los mostrados en las Figuras 3.54 y 3.55 gracias a la exacta recuperación de la singularidad. Esta situación no se produce en el procedimiento de multiplicación por (ρ) .

No obstante, este hecho no significa que la recuperación de las funciones de Green a partir de valores obtenidos con las RBFNNs que utilizan la multiplicación por (ρ) sea imprecisa. Durante el entrenamiento de las RBFNNs se fuerza a que el error NMSE caiga por debajo del umbral fijado tanto para los valores de salida de las red neuronales como para esos mismos valores con la singularidad

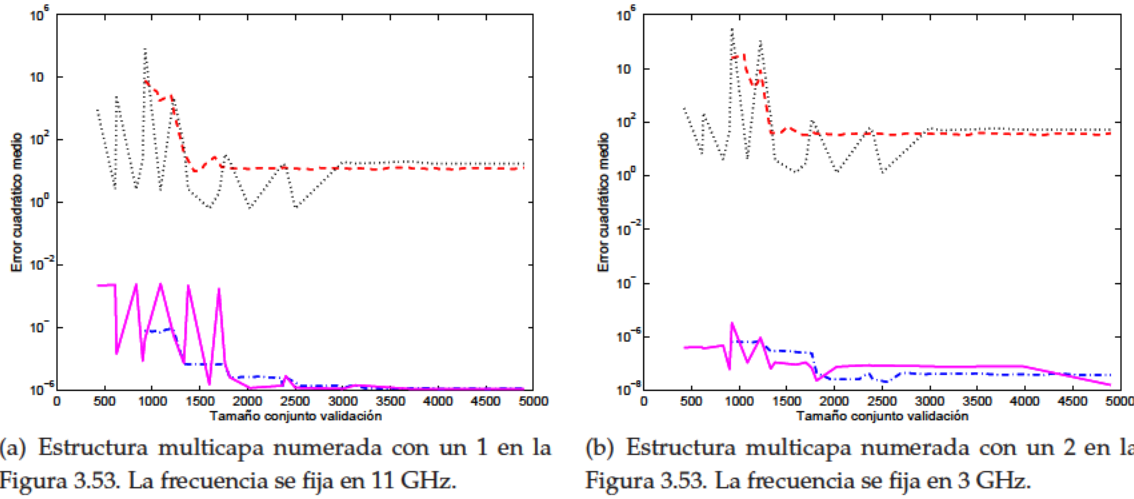


Figura 3.58: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$.

recuperada. Los valores de las funciones de Green con singularidad son mucho mayores que los valores sin singularidad. Por esta razón, el error cuadrático medio toma valores elevados respecto a la situación sin singularidad, cuando se usa la multiplicación por (ρ) . Esto da una idea clara de la gran precisión de la aproximación de las funciones de Green cerca de la fuente con la nueva técnica. La singularidad es totalmente reconstruida a partir de las capas extraídas. Cerca de las paredes la diferencia entre ambas estrategias es ligeramente menor tal como se observa en las Figuras 3.60 y 3.61. Esta circunstancia es debida a que las funciones de Green en esta zona toman valores menores que en el centro de la cavidad. Así, la suma total que se realiza para evaluar el error cuadrático medio será en general menor que en la zona alejada de las paredes. De todos modos, la nueva técnica de eliminación de la singularidad supera ampliamente el rendimiento de la estrategia de multiplicación por (ρ) también cerca de los muros metálicos, hasta alcanzar una diferencia cifrada entre seis y diez órdenes de magnitud en ganancia de precisión.

3.9.3. Análisis de circuitos encapsulados con las nuevas técnicas de aproximación de las funciones de Green

Las nuevas técnicas de aproximación de las funciones de Green han sido aplicadas en dos cavidades distintas. Con el objetivo de comparar el rendimiento de los distintos algoritmos desarrollados, la primera cavidad empleada, numerada con un 1 en la Figura 3.53, es la misma que la utilizada en la Sección 3.8 (ver Figuras 3.37 y 3.38). Para ampliar el test de las nuevas técnicas, un filtro de tipo horquilla ha sido analizado en la segunda cavidad de la Figura 3.53. Además, el ancho de banda de trabajo empleado en la segunda caja es diferente del utilizado en la primera estructura. Por tanto, la segunda cavidad constituye un caso totalmente distinto al estudiado previamente en la cavidad primera.

En primer lugar, las funciones de Green en ambas cavidades han sido aproximadas utilizando

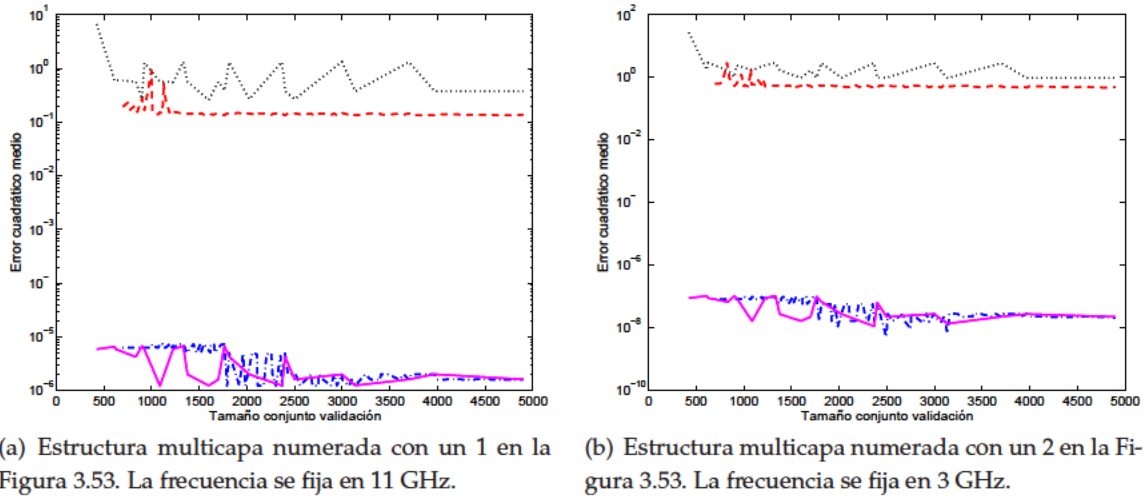


Figura 3.59: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$.

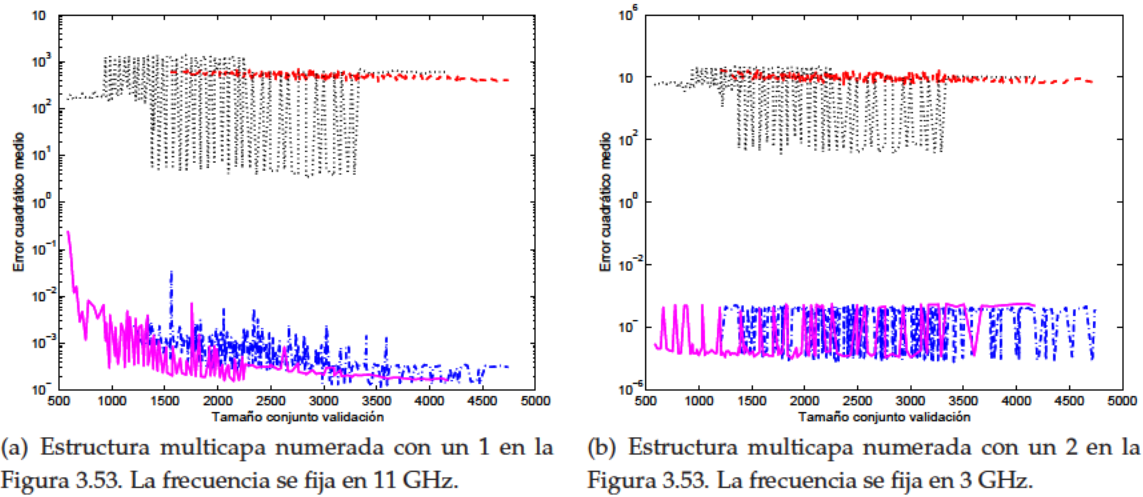
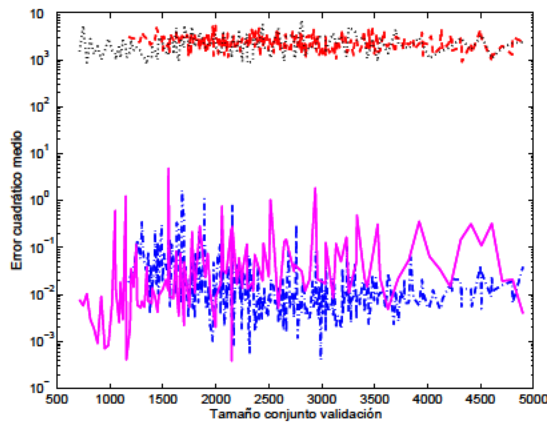


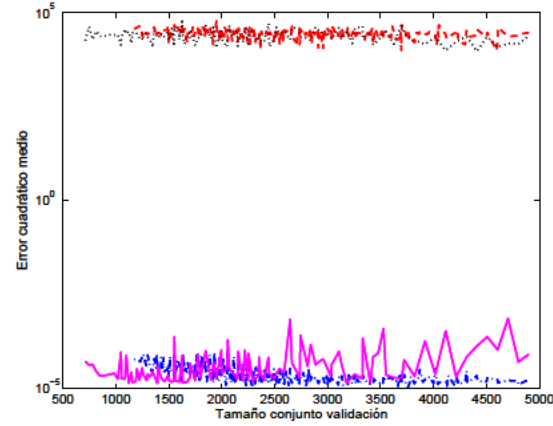
Figura 3.60: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$.

la técnica de extracción de dos capas de influencia combinada con zonas circulares en la región MS, es decir, cerca de la fuente. Con el propósito de comparar las dos estrategias de eliminación de la singularidad, el método de multiplicación por la distancia (ρ) también ha sido combinado con las zonas de tipo circular cerca de la fuente. Para completar la comparativa, la técnica desarrollada en la Sección 3.8 basada en la multiplicación por la distancia (ρ) combinada con zonas cuadradas ha sido utilizada, asimismo, en las dos cavidades bajo estudio. En todos los casos se ha empleado el mecanismo de varianza adaptativa descrito en la Figura 3.42 para calcular las zonas observación.

Las zonas observación circulares y la técnica de extracción de las dos capas de influencia están limitadas a la zona MS. Esta zona está delimitada por el umbral $U_x = U_y$. Según se explicó en la



(a) Estructura multicapa numerada con un 1 en la Figura 3.53. La frecuencia se fija en 11 GHz.



(b) Estructura multicapa numerada con un 2 en la Figura 3.53. La frecuencia se fija en 3 GHz.

Figura 3.61: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$.

Sección 2.4, este umbral diferencia la región en la que se aplica el método de la suma de imágenes para evaluar las funciones de Green de la región de débil singularidad en la que se emplea la técnica de suma por partes. Como se mostró en la Sección 2.4 dicho límite se obtiene de forma experimental. Las Figuras 2.12 y 2.13 permiten deducir que el umbral en la cavidad 1 de la Figura 3.53 debe fijarse a un valor aproximadamente de $U_x = 0,20/\pi$ (ver ecuaciones 2.75 y 2.76 y los resultados mostrados en las Figuras 2.12 y 2.13). En consecuencia, aquellos puntos observación que cumplan el siguiente par de condiciones: $|x - x'|/a \leq 0,20/\pi$ y $|y - y'|/b \leq 0,20/\pi$, pertenecerán a la zona MS. La determinación correcta del umbral de la zona MS, posibilita la minimización del tiempo de evaluación de las funciones de Green. Así, el tiempo de generación de los conjuntos de entrenamiento y validación es a su vez minimizado de modo que la comparación temporal entre la solución directa y el método neuronal es más acertada. Fuera de la región MS las zonas observación toman una forma cuadrada y son multiplicadas por (ρ) para conseguir si cabe un comportamiento menos abrupto. Gracias a la nueva técnica de eliminación de la singularidad el error NMSE máximo de aproximación cerca de la fuente ha podido ser disminuido hasta un valor tan bajo como 0,025. Al contrario de lo que ocurría con la antigua técnica de cancelación de la singularidad, el error cerca de la fuente será menor que los niveles de error encontrados en las zonas más alejadas. Por consiguiente, no se puede fijar un valor de error máximo común a todas las zonas observación de valor igual al error máximo alcanzado en la primera zona observación tal como se realizó en [Garcia et al., 2006b]. Además, la precisión en la aproximación de las funciones de Green en la región DS mejora conforme nos alejamos de la fuente. De este modo, es aconsejable permitir diferentes niveles máximos de error en la región DS, dependiendo de la cercanía a la fuente, al igual que en la aplicación del algoritmo de división efectuada en la Sección 3.8.

En la Tabla 3.15 se observan las zonas observación generadas en la región MS cuando se extraen dos capas y se emplean zonas circulares. Tan solo son necesarias cuatro zonas, más la zona de transición, para conseguir un error NMSE máximo común de 0,025. En la región DS el número de zonas necesarias para lograr la precisión requerida en cada caso es superior al obtenido en la región MS,

como se aprecia en la Tabla 3.16. El algoritmo de varianza adaptativa asigna un valor común de varianza espacial $\sigma_s^2 = 0,050$ a todas las zonas observación de la región MS. Este hecho es debido a la sencillez de la aproximación gracias a la nueva técnica de extracción de la singularidad.

Tabla 3.15: Zonas observación circulares con extracción de dos capas situadas en la región MS definida como: $0,0 \leq (|x - x'|/a, |y - y'|/b) \leq 0,20/\pi$. Cavidad 1 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	Posición del punto fuente	$0,050/\pi$	0.050
2ª zona	$0,050/\pi$	$0,125/\pi$	0.050
3ª zona	$0,125/\pi$	$0,162/\pi$	0.050
4ª zona	$0,162/\pi$	$0,200/\pi$	0.050

Tabla 3.16: Zonas observación cuadradas con multiplicación por (ρ) en la región DS. Cavidad 1 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	$0,200/\pi$	$0,250/\pi$	0.050
2ª zona	$0,250/\pi$	$0,300/\pi$	0.050
3ª zona	$0,300/\pi$	$0,375/\pi$	0.050
4ª zona	$0,375/\pi$	$0,600/\pi$	0.025
5ª zona	$0,600/\pi$	1,000	0.050

Los puntos fuente del primer cuadrante asociados a cada una de las zonas observación son sometidos a un proceso de división. En la Tabla 3.17 se muestran los límites de cada una de las zonas fuente obtenidas para las zonas observación de la región MS. La mejora posibilitada con las nuevas técnicas es tal, que el número mínimo de zonas fuente se consigue con la zona de puntos observación más cercanos a la fuente. En las zonas intermedias de la región MS y de la región DS (Figura 3.18) es necesario una división más intensa para expresar correctamente las variaciones de las funciones de Green. La extensión del algoritmo de división de zonas fuente de la Sección 3.36 prueba así su utilidad.

Cabe destacar que el tiempo total invertido en la selección de las zonas y en el entrenamiento con las nuevas técnicas es menor que el requerido en el algoritmo desarrollado en la Sección 3.8 (comparar Tablas 3.19 y 3.13). Dicha disminución es conseguida a pesar de que el nuevo algoritmo realiza una división del espacio de entrada más profunda. La reducción del tiempo invertido es debido a dos causas. En primer lugar, el algoritmo OLS mejorado permite una importante reducción del tiempo de entrenamiento. En segundo lugar, gracias a la técnica de extracción de dos capas, el entrenamiento es más sencillo y por tanto el número de neuronas requeridas para alcanzar el error mínimo será menor, incluso si el nivel de NMSE máximo ha bajado hasta 0,025. De este modo, el tiempo de entrenamiento correspondiente a las zonas observación de la región MS es disminuido. Se observa que el tiempo de selección mostrado en la Tabla 3.19 es mayor que el tiempo empleado cuando se utiliza la técnica de multiplicación por (ρ) combinada con zonas cuadradas (ver Tabla 3.13). Este hecho se debe a que el espacio de entrada es dividido ahora más intensamente, sobre todo en la parte correspondiente a los puntos fuente. Si únicamente se realizase una división de los puntos

Tabla 3.17: Zonas fuente correspondientes a las zonas observación circulares con extracción de dos capas de la región MS. Zona de transición de la región MS con multiplicación por (ρ). Cavidad 1 de la Figura 3.53.

Zona	1ª zona observ.		2ª zona observ.		3ª zona observ.		4ª zona observ.		5ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000	0.3500	0.5000	0.3500	0.5000
2ª zona	0.1667	0.3000	0.1667	0.3000	0.2000	0.3000	0.2500	0.3500	0.2500	0.3500
3ª zona	0.0333	0.1667	0.0333	0.1667	0.1437	0.2000	0.1500	0.2500	0.1750	0.2500
4ª zona	0.0010	0.0333	0.0010	0.0333	0.1120	0.1437	0.0937	0.1500	0.1328	0.1750
5ª zona					0.0987	0.1120	0.0700	0.0937	0.1150	0.1328
6ª zona					0.0853	0.0987	0.0567	0.0700	0.1017	0.1150
7ª zona					0.0720	0.0853	0.0434	0.0567	0.0883	0.1017
8ª zona					0.0619	0.0720	0.0356	0.0434	0.0750	0.0883
9ª zona					0.0519	0.0619	0.0278	0.0356	0.0675	0.0750
10ª zona					0.0419	0.0519	0.0200	0.0278	0.0600	0.0675
11ª zona					0.0319	0.0419	0.0122	0.0200	0.0524	0.0600
12ª zona					0.0218	0.0319	0.0044	0.0122	0.0449	0.0524
13ª zona					0.0117	0.0218	0.0010	0.0044	0.0374	0.0449
14ª zona					0.0016	0.0117			0.0299	0.0374
15ª zona					0.0010	0.0016			0.0224	0.0299
16ª zona									0.0149	0.0224
17ª zona									0.0074	0.0149
18ª zona									0.0010	0.0074

fuelle común para todas las zonas observación, como en la Sección 3.8, el tiempo de selección sería menor que el expuesto en la Tabla 3.13 gracias al uso del algoritmo OLS mejorado. Por lo tanto, las nuevas técnicas posibilitan una drástica disminución del error NMSE reduciendo al mismo tiempo el periodo de selección y entrenamiento.

La estrategia de extracción de la singularidad consistente en la multiplicación por la distancia (ρ) es combinada con zonas circulares para generar las zonas observación de la Tabla 3.20. El uso de las zonas circulares permite un ligero incremento de la precisión respecto al empleo de zonas cuadradas de la Sección 3.8. Así, el mínimo error NMSE alcanzado por las zonas de puntos observación más cercanos a la fuente es igual a 0,09, de todas formas, mayor que el conseguido con la nueva técnica de extracción. El descenso en el error máximo cerca de la fuente fuerza una división de las zonas fuente más complicada que la realizada en la Sección 3.8. En esta ocasión, la división de las zonas fuente realizada con la primera zona observación no es aplicada al resto de zonas observación. Conforme los puntos observación se alejan de la fuente, la aproximación se torna más sencilla y por tanto son necesarias menos zonas fuente. Este hecho provoca una disminución del tiempo de entrenamiento total requerido para obtener las RBFNNs, para estas zonas observación, en comparación con los resultados de la Sección 3.8. Sin embargo, el cálculo de las anteriores zonas fuente incrementa el tiempo de división del espacio de entrada. Debido a la mayor dificultad en la aproximación de las

Tabla 3.18: Zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) en la región DS. Cavidad 1 de la Figura 3.53.

Zona	1ª zona observ.		2ª zona observ.		3ª zona observ.		4ª zona observ.		5ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000	0.3000	0.5000
2ª zona	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000	0.1667	0.3000
3ª zona	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333	0.0333	0.1667
4ª zona	0.2000	0.2667	0.2000	0.2667	0.2000	0.2667	0.2292	0.2667	0.0151	0.0333
5ª zona	0.1500	0.2000	0.1467	0.2000	0.1500	0.2000	0.2081	0.2292	0.0092	0.0151
6ª zona	0.1219	0.1500	0.1309	0.1467	0.1125	0.1500	0.1923	0.2081	0.0067	0.0092
7ª zona	0.1008	0.1219	0.1150	0.1309	0.0844	0.1125	0.1764	0.1923	0.0041	0.0067
8ª zona	0.0797	0.1008	0.0992	0.1150	0.0563	0.0844	0.1606	0.1764	0.0027	0.0041
9ª zona	0.0586	0.0797	0.0834	0.0992	0.0281	0.0563	0.1448	0.1606	0.0010	0.0027
10ª zona	0.0375	0.0586	0.0676	0.0834	0.0010	0.0281	0.1290	0.1448		
11ª zona	0.0164	0.0375	0.0518	0.0676			0.1132	0.1290		
12ª zona	0.0010	0.0164	0.0359	0.0518			0.0974	0.1132		
13ª zona			0.0200	0.0359			0.0816	0.0974		
14ª zona			0.0041	0.0200			0.0658	0.0816		
15ª zona			0.0010	0.0041			0.0500	0.0658		
16ª zona							0.0342	0.0500		
17ª zona							0.0184	0.0342		
18ª zona							0.0026	0.0184		
19ª zona							0.0010	0.0026		

Tabla 3.19: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Nueva técnica de extracción y zonas circulares en la región MS. Multiplicación por (ρ) y zonas cuadradas en la región DS.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	7h 30m	41h 15m	40m	12h 30m	61h 55m

funciones de Green, el número de zonas observación y fuente es mayor que el número alcanzado con la nueva estrategia de cancelación de la singularidad. Por lo tanto, incluso si se emplea el algoritmo OLS mejorado, el tiempo de entrenamiento se incrementará fuertemente

Las zonas observación y fuente correspondientes a la región DS son las mismas que las definidas en las Tablas 3.16 y 3.18 respectivamente. El tiempo de selección y entrenamiento de todas las zonas y RBFNNs se desglosa en la Tabla 3.23. El tiempo total de diseño del conjunto completo de RBFNNs es muy superior al periodo invertido con las nuevas estrategias. Esto es debido a la gran cantidad de zonas fuente y observación requeridas, especialmente cerca de la fuente, tal y como se ve en las Tablas 3.20, 3.21 y 3.22.

Tabla 3.20: Zonas observación circulares con multiplicación por (ρ) situadas la zona MS definida como: $0,0 \leq (|x - x'|/a, |y - y'|/b) \leq 0,20/\pi$. Cavidad 1 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	Posición del punto fuente	$0,0025/\pi$	0.0250
2ª zona	$0,0025/\pi$	$0,0050/\pi$	0.050
3ª zona	$0,0050/\pi$	$0,0310/\pi$	0.050
4ª zona	$0,0310/\pi$	$0,0540/\pi$	0.050
5ª zona	$0,0540/\pi$	$0,1270/\pi$	0.050
6ª zona	$0,0127/\pi$	$0,2000/\pi$	0.050

Tabla 3.21: Tres primeras zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Las zonas fuente de la zona de transición son las mismas que las exhibidas en 3.17. Cavidad 1 de la Figura 3.53.

	1ª zona observ.		2ª zona observ.		3ª zona observ.	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3500	0.5000	0.3500	0.5000	0.3000	0.5000
2ª zona	0.2500	0.3500	0.2500	0.3500	0.1667	0.3000
3ª zona	0.1500	0.2500	0.1500	0.2500	0.0333	0.1667
4ª zona	0.0500	0.1500	0.0500	0.1500	0.0151	0.0333
5ª zona	0.0384	0.0500	0.0293	0.0500	0.0092	0.0151
6ª zona	0.0267	0.0384	0.0206	0.0293	0.0047	0.0092
7ª zona	0.0202	0.0267	0.0141	0.0206	0.0032	0.0047
8ª zona	0.0153	0.0202	0.0092	0.0141	0.0017	0.0032
9ª zona	0.0116	0.0153	0.0055	0.0092	0.0010	0.0017
10ª zona	0.0080	0.0116	0.0034	0.0055		
11ª zona	0.0059	0.0080	0.0020	0.0034		
12ª zona	0.0043	0.0059	0.0010	0.0020		
13ª zona	0.0031	0.0043				
14ª zona	0.0023	0.0031				
15ª zona	0.0016	0.0023				
16ª zona	0.0010	0.0016				

Finalmente, las redes neuronales, fruto de la utilización de la estrategia de multiplicación por (ρ) combinada con zonas cuadradas en todo el espacio, también se emplean en el análisis neuronal por motivos de comparación. Como se comentó en la Sección 3.8 las RBFNNs correspondientes a esta estrategia han sido re-entrenadas para todas las combinaciones de zonas observación y fuente de las Tablas 3.10 y 3.11. Este re-entrenamiento tiene por objeto el aumento de la precisión en aquellas RBFNNs cuyas zonas observación se encuentren alejadas de la fuente.

Para ilustrar el rendimiento de las diferentes estrategias de aproximación, los filtros prácticos definidos en las Figuras 3.37 y 3.38 han sido analizados en el ancho de banda de trabajo. En primer

Tabla 3.22: Tres últimas zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 1 de la Figura 3.53.

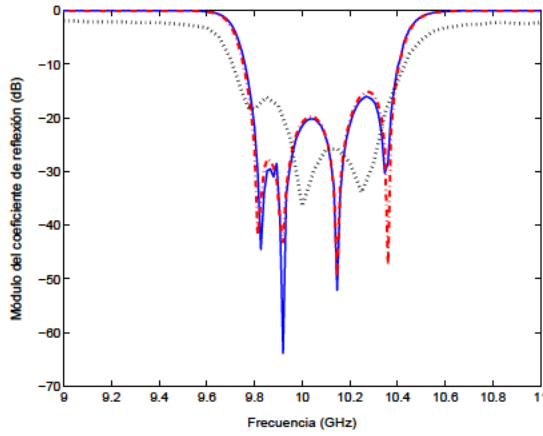
	4ª zona observ.		5ª zona observ.		6ª zona observ.	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000
2ª zona	0.1667	0.3000	0.1667	0.3000	0.1667	0.3000
3ª zona	0.1104	0.1667	0.0333	0.1667	0.0333	0.1667
4ª zona	0.0788	0.1104	0.0091	0.0333	0.0010	0.0333
5ª zona	0.0472	0.0788	0.0010	0.0091		
6ª zona	0.0156	0.0472				
7ª zona	0.0074	0.0156				
8ª zona	0.0038	0.0074				
9ª zona	0.0010	0.0038				

Tabla 3.23: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Técnica de multiplicación por (ρ) y zonas circulares en la zona MS. Técnica de multiplicación por (ρ) y zonas cuadradas en la zona DS.

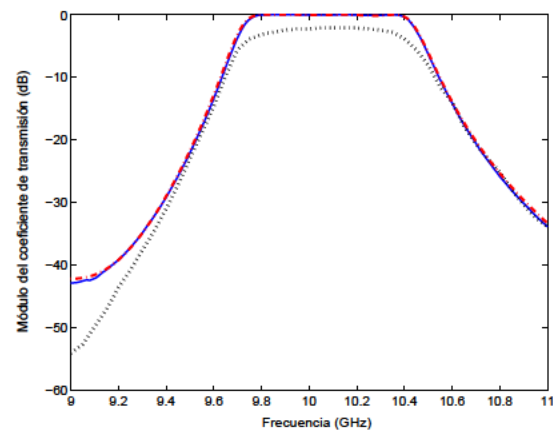
	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	10h 40m	56h 56m	1h 10m	26h 32m	95h 18m

lugar, podemos ver en las Figuras 3.62 y 3.63 los resultados correspondientes a las nuevas técnicas. En ambos filtros el método neuronal sigue con elevada precisión la evolución de los parámetros S. Las variaciones del coeficiente de reflexión (S_{11}) de la solución exacta en la parte central de la banda de paso son seguidas con gran fidelidad por el método neuronal tanto en el primer filtro, Figura 3.62(a), como en el segundo, Figura 3.63(a). La mejora respecto a la técnica de multiplicación por (ρ) y zonas cuadradas de la Sección precedente es sobresaliente. Todos los ceros de reflexión son correctamente ubicados en ambos circuitos al contrario de lo que ocurría en el método de la Sección 3.8. Las simulaciones nos enseñan que, gracias a las nuevas técnicas implantadas en el entrenamiento de las RBFNNs, el método neuronal puede ser utilizado en herramientas de diseño con gran confianza sobre la precisión conseguida.

Es interesante conocer el grado de responsabilidad de cada una de las nuevas estrategias en la mejora de la precisión del método neuronal. Para determinar dicho grado, se expone en las Figuras 3.64 y 3.65 una comparativa de los parámetros S obtenidos con las tres estrategias estudiadas. La técnica de multiplicación por (ρ) combinada con zonas circulares consigue resultados más precisos que los alcanzados con zonas observación cuadradas. En este caso, hay que atribuir en buena medida la mejora a una división más densa del espacio de entrada. La profunda división de las zonas observación en la región MS y DS, así como la división de las zonas fuentes para cada zona observación, permite que el método neuronal siga correctamente al comienzo y fin de la banda de paso las variaciones de los parámetros S. Sin embargo, a pesar de la mejoría en la simulación electromagnética

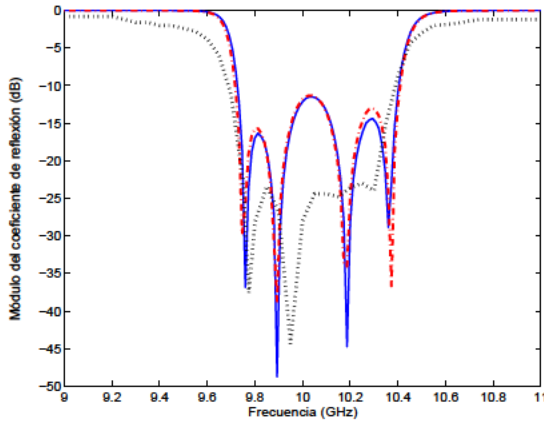


(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

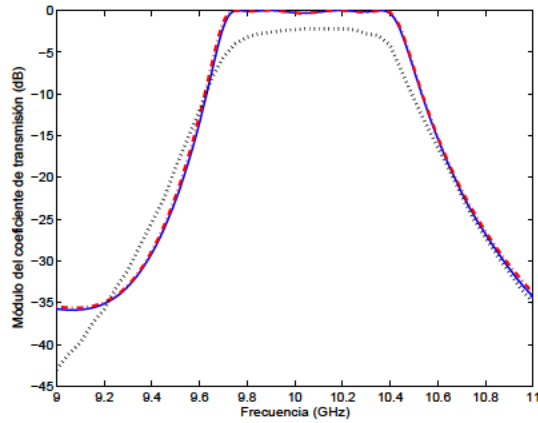


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

Figura 3.62: Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.



(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

Figura 3.63: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.

respecto al método que usa zonas cuadradas, la utilización de zonas circulares y la mayor división del espacio no logran que las variaciones en la banda de paso sean reflejadas con total verosimilitud. La utilización de zonas circulares permite un ligero incremento de la precisión como se deduce de las Figuras 3.54(a) y 3.57(b). No obstante, el uso de estas nuevas zonas no es capaz por sí solo de incrementar la calidad de la aproximación de las funciones de Green lo suficiente para llegar a la precisión deseada. Como podemos apreciar en las Figuras 3.64 y 3.65 la técnica de extracción de dos capas

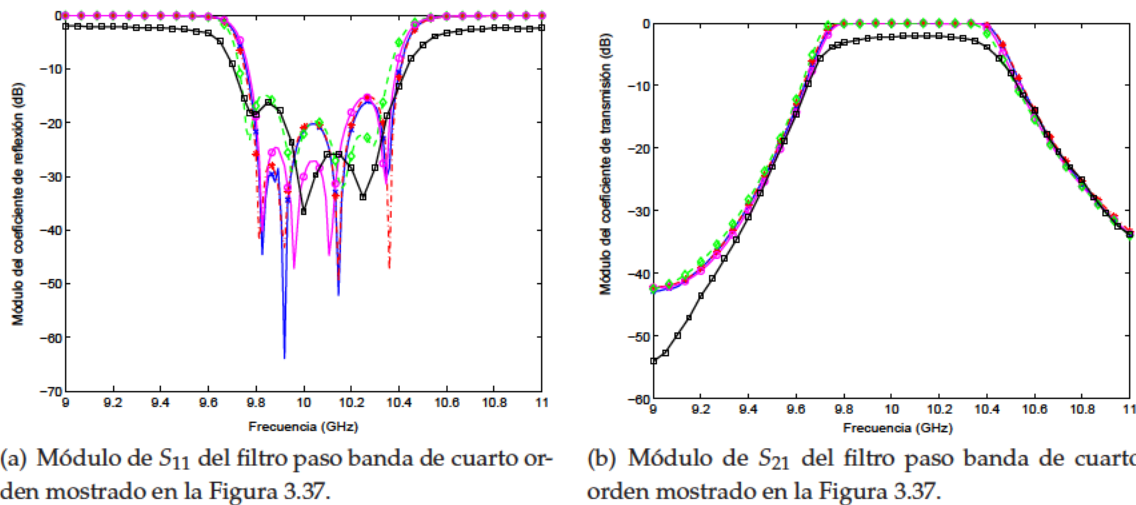


Figura 3.64: Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.37. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua, cuadrados).

consigue que el análisis neuronal alcance una elevada exactitud. Las diferencias de los parámetros S_{11} y S_{21} obtenidos con las nuevas estrategias respecto a la solución directa son mínimas en todo el ancho de banda para los dos filtros impresos. La precisión lograda es atribuible en gran medida a la eliminación de la singularidad mediante la extracción de dos capas de influencia. Las nuevas zonas circulares y la división general de las zonas fuente refuerzan el rendimiento de la nueva técnica de cancelación de la singularidad. Hay que tener en cuenta que las RBFNNs aproximan las funciones de Green desprovistas de singularidad y que ésta es recuperada de forma rigurosa durante la resolución de la IE gracias a la adición de forma exacta las dos primeras capas de influencia.

Generalmente cuanto mayor sea el acoplo entre las tiras metálicas que componen el filtro impreso mayor será la precisión requerida en la aproximación de las funciones de Green para arrojar resultados correctos. Así, con el propósito de testear el funcionamiento del método neuronal en un circuito impreso con separaciones muy pequeñas entre tiras impresas se ha procedido al análisis del filtro dibujado en la Figura 3.66. Este filtro posee una separación entre el puerto de entrada y salida con el primer y último resonador respectivamente de tan solo 0,0646 mm.

Como se aprecia en las Figuras 3.67 la técnica de extracción de dos capas de influencia, combinada con zonas circulares, alcanza gran precisión en la obtención de los parámetros S. La respuesta correspondiente al coeficiente de reflexión (S_{11}) es seguida con un grado tal de exactitud que permite la localización correcta de los cinco polos (ver Figura 3.67(a)). Asimismo, la respuesta del coeficiente de transmisión (S_{21}) es evaluada adecuadamente en todo el ancho de banda de trabajo de forma que el método neuronal logra la ubicación correcta del cero de transmisión situado al final de la banda (Figura 3.67(b)).

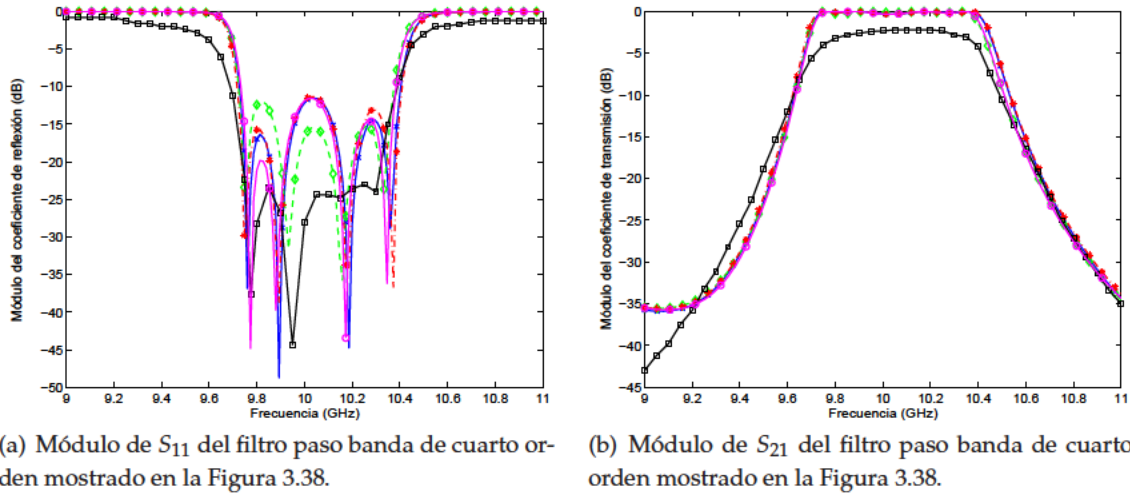


Figura 3.65: Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.38. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua, cuadrados).

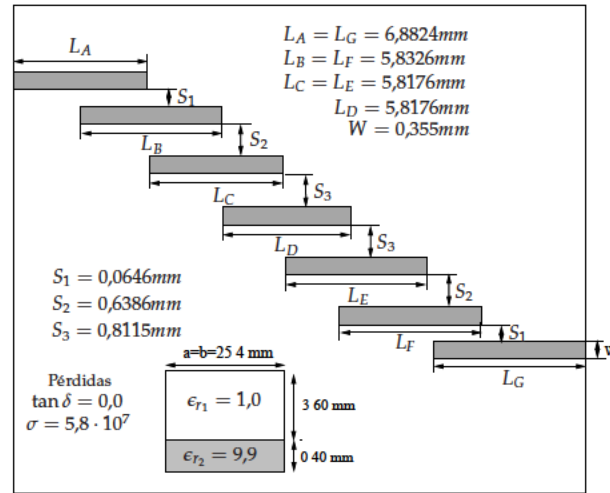
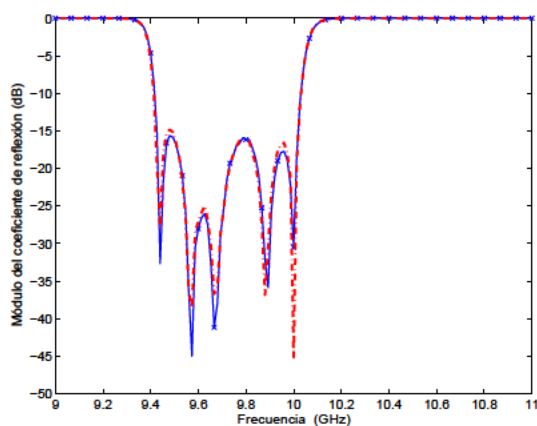
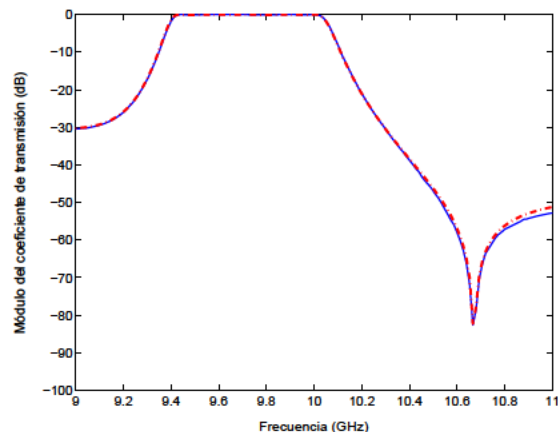


Figura 3.66: Filtro paso banda de cinco polos basado en líneas acopladas. Se incluyen las dimensiones de la estructura multicapa apantallada (Cavidad 1 de la Figura 3.53).

Los resultados obtenidos con la nueva técnica de eliminación de la singularidad superan con creces el rendimiento de las técnicas que emplean la técnica de multiplicación por (ρ) como se observa en las Figuras 3.68. Este hecho se valora especialmente en la evaluación del coeficiente de reflexión (S_{11}) de la Figura 3.68(a). La estrategia de multiplicación por (ρ) combinada con zonas circulares sigue con corrección la respuesta del filtro en la parte inicial y final de la banda de paso pero falla a lo largo de toda la zona central de dicha banda. Por otro lado, la técnica de multiplicación por (ρ) con zonas cuadradas presenta errores tanto en la parte central de la banda de paso como al principio y

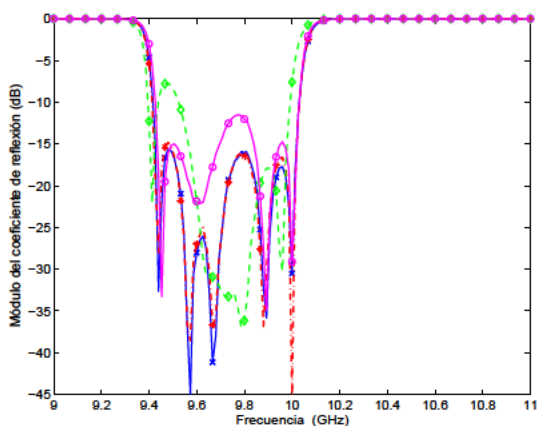


(a) Módulo de S_{11} del filtro paso banda de quinto orden mostrado en la Figura 3.66.

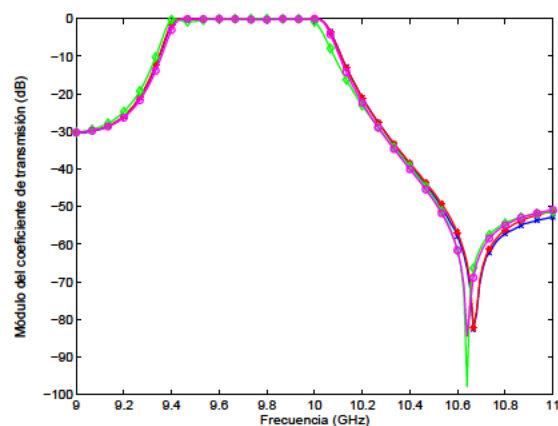


(b) Módulo de S_{21} del filtro paso banda de quinto orden mostrado en la Figura 3.66.

Figura 3.67: Respuesta del filtro de quinto orden de la Figura 3.66 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.66.



(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.66.

Figura 3.68: Comparativa entre las respuestas del filtro de quinto orden de la Figura 3.66. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes).

final de la mencionada banda. Respecto a la respuesta correspondiente al coeficiente de transmisión (S_{21}) podemos observar que la localización del cero de transmisión no es exacta cuando se utiliza la técnica de multiplicación por la distancia (ρ) ya sea combinada con zonas cuadradas como con zonas circulares (Figura 3.68(b)).

La principal razón para sustituir las funciones de Green exactas por los valores de salida de las RBFNNs entrenadas reside en la gran cantidad de tiempo ahorrado en el cálculo de la respuesta eléctrica de los circuitos tal como se ha visto en las secciones 3.7 y 3.8. En la Tabla 3.24 se exhibe una comparación entre los tiempos necesarios para analizar los filtros de cuarto y quinto orden con el método neuronal, así como con el método directo que hace uso de la evaluación numérica de las funciones de Green. El nuevo procedimiento de extracción de dos capas de influencia obtiene la respuesta más rápida en todos los casos. Este resultado se debe a que la nueva técnica de eliminación de la singularidad requiere un número menor de neuronas cerca de la fuente para alcanzar el error deseado. Esta ventaja ocurre incluso en el caso de fijar un umbral de error máximo más pequeño que el utilizado con las técnicas de multiplicación por (ρ) . Todo ello es posible gracias a que el nuevo mecanismo de eliminación de la singularidad produce una serie de funciones que pueden ser aproximadas fácilmente. De este modo, la nueva técnica de eliminación de la singularidad requiere una media de 226 neuronas (con una desviación estándar de 95 neuronas) para la primera zona observación (puntos cercanos a la fuente). Las funciones de Green de los puntos observación de la mencionada zona van a ser evaluadas con mucha frecuencia durante la resolución de la IE con el MoM. Por tanto, las RBFNNs que aproximan las funciones de Green en esta zona del espacio de entrada van a ser responsables de forma relativamente importante del tiempo de análisis. Las estrategias que utilizan la multiplicación por (ρ) necesitan un número mayor de neuronas en un área de puntos observación similar a la abarcada por la primera zona observación de la nueva técnica. Así, el procedimiento de multiplicación por (ρ) con zonas circulares necesita una media de 403 neuronas (con una desviación estándar de 291 neuronas). Además, las RBFNNs entrenadas con la antigua técnica de eliminación de la singularidad combinada con zonas cuadradas utilizan 606 neuronas de media (con una desviación estándar de 176 neuronas). Durante la resolución de la IE, las dos capas de influencia previamente extraídas deben ser sumadas a los valores de la RBFNN. Esta evaluación añade un tiempo extra al tiempo total de obtención de los valores neuronales, pero es pequeño ya que las dos primeras capas de influencia se añaden de forma analítica con una suma finita y de pocos términos. Incluso con este tiempo extra, la técnica basada en la extracción de dos capas es la más eficiente.

Todos los métodos neuronales son cerca de 200 veces más rápidos que la solución directa. Incluso con la más lenta de las técnicas estudiadas el método neuronal resulta atractivo desde el punto de vista temporal. La ganancia obtenida por los métodos neuronales permite considerar el tiempo de entrenamiento y selección como despreciable incluso si se emplea el método de multiplicación por (ρ) con zonas circulares que posee el periodo de diseño más lento. Esta consideración es especialmente cierta si se analizan o diseñan diversos circuitos en la misma cavidad multicapa.

Las funciones de Green de la segunda cavidad de la Figura 3.53 han sido aproximadas con las tres estrategias anteriores. En esta estructura el ancho de banda de trabajo se extiende desde 1 GHz hasta 3 GHz. Al igual que en la cavidad primera el umbral usado para diferenciar la región MS de la región DS es igual a $0,20/\pi$. El nivel de error máximo para el método de extracción de dos capas con zonas circulares se fijó en un valor igual a 0,025 en la región MS. En la Tabla 3.25 se observan los límites y varianza de las zonas observación en esta región. El pequeño número de zonas demuestra, también en esta cavidad, la sencillez de la aproximación de las funciones de Green cuando se emplean las nuevas técnicas. De hecho, el mecanismo de división adaptativa no necesita realizar cambios sobre la varianza ya que la aproximación es fácil una vez ha sido extraída la singularidad. En la Tabla 3.26

Tabla 3.24: Comparación en la primera cavidad de los resultados temporales de la solución directa y los tres métodos neuronales para el cálculo de los parámetros S en 151 puntos de frecuencia.

Filtro	Solución directa	Técnica de extracción de dos capas zonas circulares	Multiplicación por ρ zonas circulares	Multiplicación ρ zonas cuadradas
Cuatro polos Figura 3.37	112 h 27 m.	1810 s	2298 s	2445 s
Cuatro polos Figura 3.38	112 h 46 m.	1750 s	2171 s	2425 s
Cinco polos Figura 3.66	120h 12m.	1983 s	2380 s	2759 s

se exponen las zonas observación que cubren la región DS. En esta parte del espacio la multiplicación por la distancia (ρ) posibilita que sean necesarias seis zonas para alcanzar la precisión deseada. Hay que tener en cuenta que el umbral de error es mayor que en la región MS. Además, siguiendo el método aplicado en la primera cavidad dicho umbral es variable y depende de la distancia a la fuente. Las zonas fuente correspondientes a cada una de las zonas observación se enumeran en las Tablas 3.27 para la región MS y 3.28 y 3.29 para la región DS. En algunas zonas de la región de puntos lejanos a la fuente es necesaria una división más intensa que en la región MS. De esta forma, se vuelve a confirmar la calidad de los nuevos métodos implantados.

Tabla 3.25: Zonas observación circulares con extracción de dos capas situadas en la zona MS, definida como: $0,0 \leq (|x - x'|/a, |y - y'|/b) \leq 0,20/\pi$. Cavidad 2 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	Posición del punto fuente	$0,050/\pi$	0.050
2ª zona	$0,050/\pi$	$0,125/\pi$	0.050
3ª zona	$0,125/\pi$	$0,155/\pi$	0.050
4ª zona	$0,155/\pi$	$0,200/\pi$	0.050

Tabla 3.26: Zonas observación cuadradas con multiplicación por ρ en la zona DS. Cavidad 2 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	$0,200/\pi$	$0,292/\pi$	0.050
2ª zona	$0,292/\pi$	$0,369/\pi$	0.050
3ª zona	$0,369/\pi$	$0,427/\pi$	0.050
4ª zona	$0,427/\pi$	$0,514/\pi$	0.025
5ª zona	$0,514/\pi$	$0,600/\pi$	0.025
6ª zona	$0,600/\pi$	1,000	0.050

El procedimiento de multiplicación por (ρ) combinado con zonas circulares también ha sido

Tabla 3.27: Zonas fuente correspondientes a las zonas observación circulares con extracción de dos capas de la región MS incluyendo la zona de transición. Cavity 2 de la Figura 3.53.

Zona	1ª zona observ.		2ª zona observ.		3ª zona observ.		4ª zona observ.		5ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000
2ª zona	0.1667	0.3000	0.1667	0.3000	0.1667	0.3000	0.1667	0.3000	0.1667	0.3000
3ª zona	0.0333	0.1667	0.0667	0.1667	0.0667	0.1667	0.0917	0.1667	0.0667	0.0667
4ª zona	0.0010	0.0333	0.0010	0.0667	0.0001	0.0667	0.0354	0.0917	0.0010	0.0667
5ª zona							0.0010	0.0354		

Tabla 3.28: Tres primeras zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) de la región DS. Cavity 2 de la Figura 3.53.

Zona	1ª zona observ.		2ª zona observ.		3ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000
2ª zona	0.1667	0.3000	0.1667	0.3000	0.1667	0.3000
3ª zona	0.1245	0.1667	0.1104	0.1667	0.1245	0.1667
4ª zona	0.1007	0.1245	0.0542	0.1104	0.0823	0.1245
5ª zona	0.0830	0.1007	0.0010	0.0542	0.0401	0.0823
6ª zona	0.0652	0.0830			0.0010	0.0401
7ª zona	0.0474	0.0652				
8ª zona	0.0296	0.0474				
9ª zona	0.0118	0.0296				
10ª zona	0.0010	0.0118				

aplicado a la aproximación de las funciones de Green. El valor mínimo de NMSE alcanzado es igual a 0,085, mayor por tanto que el nivel máximo de error conseguido con las nuevas técnicas. La tarea de aproximación se torna más difícil, por consiguiente las regiones de puntos observación y fuente

Tabla 3.29: Tres últimas zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) de la región DS. Cavity 2 de la Figura 3.53.

Zona	4ª zona observ.		5ª zona observ.		6ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.3000	0.5000	0.3000	0.5000
2ª zona	0.1667	0.3000	0.2000	0.3000	0.2000	0.3000
3ª zona	0.1104	0.1667	0.1000	0.2000	0.1437	0.2000
4ª zona	0.0542	0.1104	0.0257	0.1000	0.0875	0.1437
5ª zona	0.0010	0.0542	0.0010	0.0257	0.0312	0.0875
6ª zona					0.0010	0.0312

del espacio de entrada deben ser divididos de forma más densa como se observa en las Tablas 3.30, 3.31 y 3.32.

Tabla 3.30: Zonas observación cuadradas con multiplicación por (ρ) situadas en la zona MS definida como: $0,0 \leq (|x - x'|/a, |y - y'|/b) \leq 0,20/\pi$. Cavidad 2 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1ª zona	Posición del punto fuente	0,0050/ π	0.050
2ª zona	0,0050/ π	0,0880/ π	0.050
3ª zona	0,0880/ π	0,0125/ π	0.050
4ª zona	0,0125/ π	0,0250/ π	0.050
5ª zona	0,0250/ π	0,0456/ π	0.050
6ª zona	0,0456/ π	0,0817/ π	0.050
7ª zona	0,0817/ π	0,2000/ π	0.050

Tabla 3.31: Cinco primeras zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.

Zona	1ª zona observ.		2ª zona observ.		3ª zona observ.		4ª zona observ.		5ª zona observ.	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000
2ª zona	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000
3ª zona	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333
4ª zona	0.2000	0.2667	0.2000	0.2667	0.2000	0.2667	0.2000	0.2667	0.2000	0.2667
5ª zona	0.1333	0.2000	0.1333	0.2000	0.1333	0.2000	0.1333	0.2000	0.1333	0.2000
6ª zona	0.0833	0.1333	0.0667	0.1333	0.0667	0.1333	0.0667	0.1333	0.0667	0.1333
7ª zona	0.0458	0.0833	0.0297	0.0667	0.0297	0.0667	0.0297	0.0667	0.0297	0.0667
8ª zona	0.0300	0.0458	0.0136	0.0297	0.0136	0.0297	0.0136	0.0297	0.0136	0.0297
9ª zona	0.0142	0.0300	0.0065	0.0136	0.0065	0.0136	0.0010	0.0136	0.0041	0.0136
10ª zona	0.0068	0.0142	0.0034	0.0065	0.0010	0.0065			0.0010	0.0041
11ª zona	0.0035	0.0068	0.0020	0.0034						
12ª zona	0.0021	0.0035	0.0010	0.0020						
13ª zona	0.0013	0.0021								
14ª zona	0.0010	0.0013								

Por último, la antigua estrategia de extracción de la singularidad combinada con zonas cuadradas ha sido aplicada en esta segunda cavidad. Por motivos comparativos, las zonas fuente son calculadas haciendo uso únicamente de la primera zona observación. El resto de zonas observación adquieren las mismas zonas fuente que aquellas obtenidas con la zona de puntos observación más cercanos a la fuente. El error NMSE máximo logrado para la primera zona observación es de 0,10. Las zonas observación se desglosan en la Tabla 3.33 mientras que las zonas fuente se muestran en la Tabla 3.34. Siguiendo la metodología de las secciones 3.8 y 3.9, durante el entrenamiento se han permitido menores niveles de error, que el conseguido en la primera zona observación, en aquellas

Tabla 3.32: Tres últimas zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.

	6 ^a zona observ.		7 ^a zona observ.		8 ^a zona observ.	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1 ^a zona	0.4000	0.5000	0.4000	0.5000	0.4000	0.5000
2 ^a zona	0.3333	0.4000	0.3333	0.4000	0.3333	0.4000
3 ^a zona	0.2667	0.3333	0.2667	0.3333	0.2667	0.3333
4 ^a zona	0.2000	0.2667	0.2000	0.2667	0.2000	0.2667
5 ^a zona	0.1333	0.2000	0.1333	0.2000	0.1333	0.2000
6 ^a zona	0.0833	0.1333	0.0667	0.1333	0.0667	0.1333
7 ^a zona	0.0458	0.0833	0.0459	0.0667	0.0303	0.0667
8 ^a zona	0.0247	0.0458	0.0303	0.0459	0.0147	0.0303
9 ^a zona	0.0089	0.0247	0.0147	0.0303	0.0010	0.0147
10 ^a zona	0.0010	0.0089	0.0044	0.0147		
11 ^a zona			0.0010	0.0044		

zonas observación situadas lejos de la fuente.

Tabla 3.33: Zonas observación cuadradas con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.

Zona	Límite inferior	Límite superior	Varianza σ^2
1 ^a zona	Posición del punto fuente	0,0070	0.010
2 ^a zona	0,0070	0,0125	0.050
3 ^a zona	0,0125	0,0200	0.050
4 ^a zona	0,0200	0,0450	0.050
5 ^a zona	0,0450	0,0900	0.050
6 ^a zona	0,0900	0,1500	0.050
7 ^a zona	0,1500	0,2000	0.050
8 ^a zona	0,2000	0,2750	0.050
9 ^a zona	0,2750	0,1000	0.050

El periodo inicial invertido en la definición de las zonas del espacio de entrada y en el entrenamiento de las RBFNNs, cuando se emplea el nuevo procedimiento de eliminación de la singularidad, se muestra en la Tabla 3.35. El tiempo total es menor que el tiempo necesitado en la primera cavidad (Tabla 3.19) debido a que las variaciones de las funciones de Green son más rápidas en cajas eléctricamente más grandes. En la Tabla 3.36 se expone el tiempo de selección y entrenamiento de la estrategia que combina la multiplicación por (ρ) con las zonas observación circulares. Tanto el tiempo de selección de las zonas observación, fuente y frecuencia como el tiempo invertido en el entrenamiento son mayores que los correspondientes tiempos de la técnica de extracción de dos capas. La diferencia temporal se debe atribuir a los distintos tiempos de selección y entrenamiento en la región MS. En la región DS ambas estrategias comparten las mismas zonas y por tanto arrojan resultados temporales iguales. La selección de las cinco zonas observación de la región MS con la

Tabla 3.34: Zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.

Zona	Límite inferior	Límite superior
1ª zona	0.4000	0.5000
2ª zona	0.3333	0.4000
3ª zona	0.2667	0.3333
4ª zona	0.2000	0.2667
5ª zona	0.1333	0.2000
6ª zona	0.0833	0.1333
7ª zona	0.0552	0.0833
8ª zona	0.0341	0.0552
9ª zona	0.0183	0.0341
10ª zona	0.0250	0.0183
11ª zona	0.0010	0.0250

nueva estrategia es muy sencilla. En cambio con el procedimiento antiguo, a pesar de utilizar zonas circulares, son necesarias dos zonas observación más, (ver Tabla 3.30). Además, el número de zonas fuente en la zona MS cuando se emplea la antigua técnica de eliminación de la singularidad es muy superior al número de zonas fuente necesarias con la nueva técnica. Solamente en la primera zona observación son requeridas catorce zonas fuente con la antigua estrategia, por únicamente cuatro con la nueva técnica. El método de multiplicación por (ρ) implica una mayor dificultad en la aproximación de las funciones de Green en la región MS, de modo que es necesario un tiempo total dos veces superior al tiempo total mostrado en la Tabla 3.35. Tanto con la nueva estrategia de eliminación de la singularidad como con las técnicas de multiplicación por (ρ) se generó una única zona frecuencial que ocupa todo el ancho de banda de trabajo.

Tabla 3.35: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Nueva técnica de extracción y zonas circulares en la zona MS. Multiplicación por (ρ) y zonas cuadradas en la zona DS.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	2h 54m	30h 36m	37m	13h 16m	47h 23m

Tabla 3.36: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Técnica de multiplicación por (ρ) y zonas circulares en la zona MS. Técnica de multiplicación por (ρ) y zonas cuadradas en la zona DS.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento RBFNN	Tiempo total
Tiempo	12h 20m	59h 6m	50m	32h 58m	105h 14m

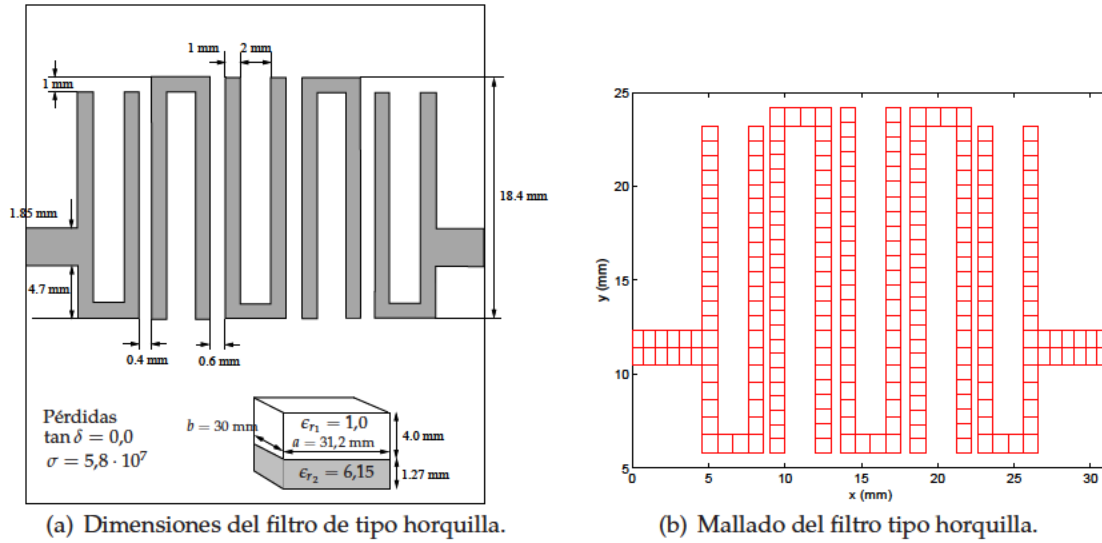
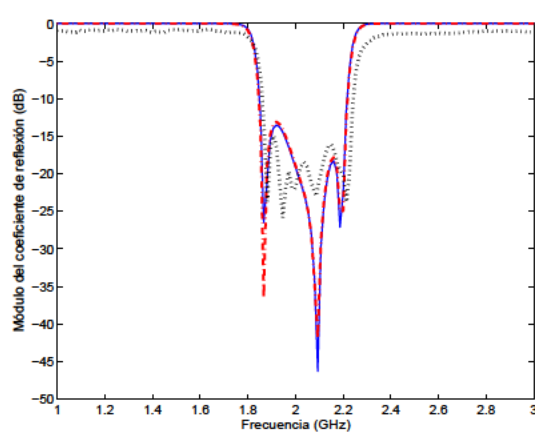


Figura 3.69: Filtro de quinto orden de tipo horquilla. Se incluyen la estructura del filtro y las dimensiones de la caja (cavidad 2 de la Figura 3.53).

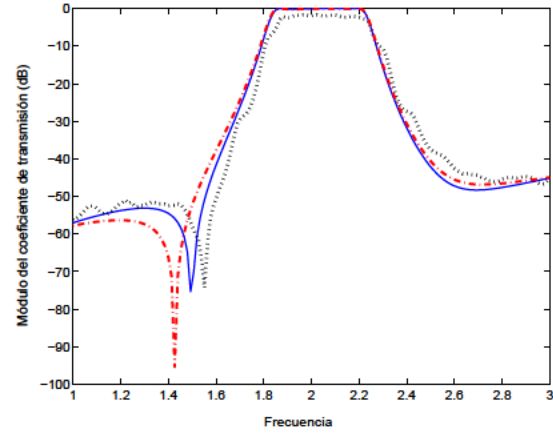
Con el objetivo de comparar la calidad de las estrategias neuronales, un filtro de tipo horquilla ha sido analizado en la segunda cavidad. Este circuito impreso es un filtro paso banda de quinto orden que se encuentra diseñado y analizado en [Hong y Lancaster, 2001]. La estructura multicapa apantallada, las dimensiones del filtro y el mallado utilizado se exhiben en la Figura 3.69.

Los parámetros S calculados con el nuevo método neuronal y con el método directo se muestran en la Figura 3.70. Además, se exponen las medidas del prototipo fabricado (ver Figura 3.71) para validar las respuestas simuladas. La evolución del parámetro S_{11} es seguida con exactitud en todo el ancho de banda 3.70(a). La localización de los polos, así como el decaimiento del parámetro S_{11} , en el inicio y final de la banda de paso es muy precisa. La respuesta del parámetro S_{21} es asimismo calculada de forma correcta por el nuevo método neuronal. Aunque existe cierto desplazamiento en la localización del cero de transmisión, los resultados son satisfactorios en todo el ancho de banda de trabajo.

En las Figuras 3.72 se presenta la comparación entre las respuestas obtenidas con las tres estrategias neuronales. En este caso, todos los métodos neuronales siguen con precisión las variaciones de la solución directa. Este filtro no presenta acoplos especialmente críticos entre las tiras metálicas impresas que lo componen. Por lo tanto, incluso el método neuronal que emplea la estrategia desarrollada en la Sección 3.8 predice la respuesta del filtro con cierta precisión. En cualquier caso, la nueva técnica de extracción combinada con las zonas circulares es la más exacta. La media de la diferencia entre la respuesta del coeficiente de reflexión (S_{11}) correspondiente a la solución directa y a la nueva técnica es igual a 0,2355 dB. La media de la diferencia del coeficiente de reflexión (S_{11}) correspondiente al mecanismo de multiplicación por (ρ) con zonas circulares sube hasta un valor de 0,4321 dB. Finalmente, la media de la diferencia correspondiente al procedimiento de multiplicación por (ρ) combinado con zonas cuadradas alcanza un valor de 0,5227 dB. Con el coeficiente de transmisión (S_{21}) se obtienen resultados análogos.



(a) Módulo de S_{11} del filtro paso banda de quinto orden mostrado en la Figura 3.69.



(b) Módulo de S_{21} del filtro paso banda de quinto orden mostrado en la Figura 3.69.

Figura 3.70: Respuesta del filtro de quinto orden de tipo horquilla de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar las simulaciones se exponen las medidas (línea punteada). La respuesta se simula para 151 puntos de frecuencia.

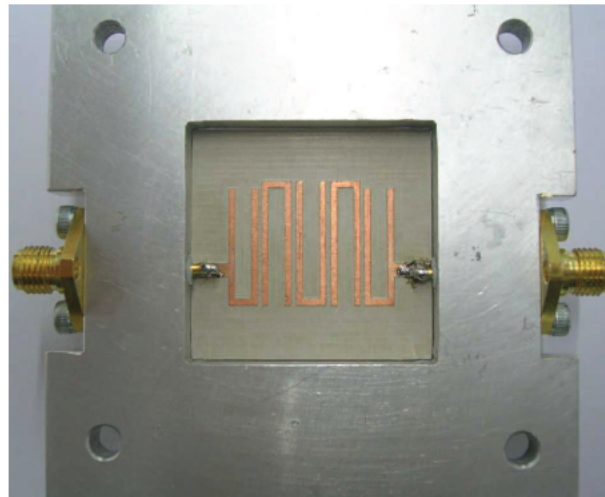


Figura 3.71: Fotografía del filtro de cinco polos de tipo horquilla fabricado.

Los métodos neuronales logran una reducción drástica del tiempo de análisis del filtro paso banda como se observa en la Tabla 3.37. Al igual que en la primera cavidad, la técnica de extracción de dos capas con zonas circulares es la estrategia más rápida en calcular la respuesta del filtro. Gracias al suave comportamiento de las funciones de Green, una vez extraídas dos capas de influencia el nuevo método neuronal necesita menos neuronas para alcanzar la precisión deseada. Así, en la primera zona observación la nueva estrategia requiere 326 neuronas de media con una desviación estándar de 95 neuronas. La suma de las cinco primeras zonas observación del método de multiplicación por (ρ) con zonas circulares cubre un área similar al área de la primera zona observación anterior. Las RBFNNs de las mencionadas cinco zonas contienen 441 neuronas de media con una desviación

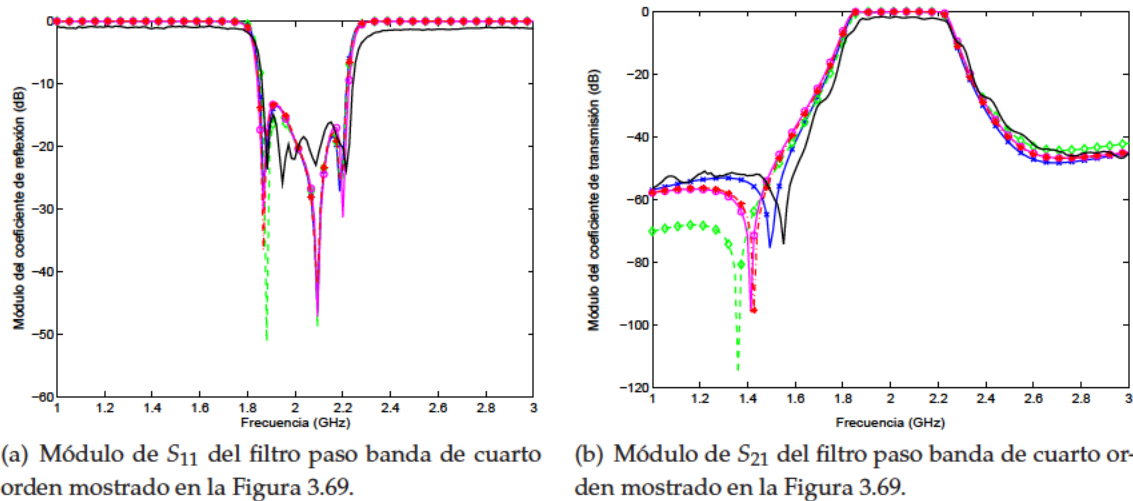


Figura 3.72: Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.69. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua).

estándar de 190 neuronas. Las tres primeras zonas de la técnica de multiplicación por (ρ) con zonas cuadradas poseen un área conjunta similar al área de la primera zona observación. En estas tres zonas son necesarias 467 neuronas con una desviación estándar de 214 neuronas. En el filtro de tipo horquilla todos los métodos neuronales poseen una precisión suficiente para su utilización como herramientas de análisis y diseño circuital eficaces. No obstante, la nueva técnica de eliminación de la singularidad es preferida ya que arroja los mejores resultados temporales y es por tanto la técnica más eficiente.

Tabla 3.37: Comparación de los resultados temporales de la solución directa y los tres métodos neuronales para el cálculo de los parámetros S en 151 puntos de frecuencia. Segunda cavidad.

Filtro	Solución directa	Técnica de extracción de dos capas zonas circulares	Multiplicación por (ρ) zonas circulares	Multiplicación por (ρ) zonas cuadradas
Cinco polos horquilla Figura 3.69	114h. 18m.	2074 s	2259 s	2152 s

3.10. Conclusiones

En el presente Capítulo se ha mostrado la técnica de análisis de circuitos encapsulados multicapa basada en redes neuronales. Después del estudio de diferentes tipos de redes neuronales se

llegó a la conclusión de que las Redes neuronales de funciones de base radial (RBFNN) son las más apropiadas para conseguir la aproximación eficiente de las funciones de Green del medio apantallado multicapa. Para respaldar la decisión tomada se realizaron diversos tests numéricos en una estructura práctica. Cada red neuronal puede ser diseñada mediante diversos métodos de entrenamiento. En este Capítulo se muestran tres de los algoritmos más utilizados en el diseño de RBFNNs. Después del análisis numérico y cualitativo de los tres algoritmos se seleccionó el método de Mínimos cuadrados ortogonales (OLS) para entrenar las RBFNNs necesarias. Este algoritmo es de tipo supervisado y permite alcanzar la precisión requerida en el complejo problema de la aproximación de las funciones de Green.

Existen principalmente dos obstáculos en la aproximación de las mencionadas funciones. Por un lado, la singularidad de la fuente produce un crecimiento muy rápido de las funciones de Green en las cercanías de la fuente. Esta característica impide la aplicación práctica de la aproximación con redes neuronales. Por otra parte, el espacio de entrada es muy extenso, de modo que el conjunto de entrenamiento para representar correctamente las funciones de Green puede alcanzar un tamaño que imposibilite el entrenamiento supervisado de la RBFNN. Para salvar los anteriores obstáculos se han empleado distintas técnicas.

Para eliminar la singularidad en la fuente se han desarrollado dos procedimientos. El primero se fundamenta en la multiplicación de las funciones de Green por la distancia entre fuente y punto observación (ρ). La segunda técnica es mucho más eficaz y consiste en la extracción de dos capas de influencia (k) compuestas por grupos de imágenes espaciales. Este procedimiento posee dos claras ventajas frente a la primera técnica. En primer lugar, permite una aproximación mucho más precisa de las funciones de Green sin singularidad. En segundo lugar, la recuperación de la singularidad es exacta. En cambio, en la técnica de multiplicación por (ρ) se produce una amplificación del error de aproximación cometido por la RBFNN cuando se recupera la singularidad. Los tests numéricos muestran que la diferencia en el error de aproximación se encuentra entre los seis y diez órdenes de magnitud.

Con el objetivo de limitar el conjunto de entrenamiento de las RBFNNs se aplicó el principio de dividir y vencer al problema de aproximación. Esta limitación evita la aparición de problemas de saturación de la memoria del ordenador. Además, posibilita el diseño de redes neuronales más compactas y precisas. En nuestro problema, es de vital interés la limitación del número de neuronas ya que cuanto menor sea el tamaño neuronal mayor será la aceleración del análisis. Por ello, el espacio de entrada compuesto por puntos fuente, puntos observación y puntos frecuencia es dividido en zonas. Los puntos observación se agrupan dependiendo de su cercanía a la fuente. Los puntos fuente se agrupan dependiendo de su ubicación respecto a las paredes metálicas. Finalmente, la coordenada frecuencial se divide en función de la aparición de frecuencias de resonancia en el ancho de banda de trabajo. Para cada combinación de zona observación, fuente y frecuencia existirá una RBFNN especializada en esa parte del espacio. Se han desarrollado una serie de algoritmos que posibilitan la completa división del espacio de entrada de forma automática. Tan solo es necesaria la intervención del diseñador para la determinación de una serie de parámetros iniciales. Como se ha podido comprobar en los distintos estudios numéricos, la varianza es un valor fundamental en la adecuada aproximación de las funciones objetivo con RBFNNs gaussianas. Este parámetro suele ser ajustado de forma heurística en muchas aplicaciones de las RBFNNs. Para alcanzar un mejor rendimiento se

ha implantado un mecanismo, que en combinación con el algoritmo de división de las zonas observación, busca de modo iterativo un valor de varianza apropiado. Como muestran los resultados obtenidos con el análisis de filtros encapsulados esta combinación de técnicas logra mejores resultados que las primeras técnicas desarrolladas.

La incidencia de la varianza en la aproximación con RBFNNs solo se vuelve importante cuando la función objetivo es de complicada modelización. Muchas de las aplicaciones con RBFNNs no toman en cuenta el ajuste de este parámetro debido a la facilidad de la aproximación. Así, estos trabajos se centran en el cálculo de los pesos de la capa lineal o de la selección de los centros a partir de los datos de entrenamiento. Todavía existe un campo amplio de investigación sobre la influencia de la varianza en aquellas redes neuronales que usan gaussianas como funciones neuronales. Por ejemplo, en el caso de las SVM el proceso de optimización cuadrática se produce sobre los pesos y centros pero no tiene en cuenta la varianza.

Cuando la extracción de la singularidad no es perfecta se ha mostrado que la profunda división del espacio de entrada y la asignación de varianzas adecuadas permite el análisis de circuitos con acoplos críticos. Sin embargo, si se desea mejorar la exactitud del análisis es necesario aplicar la técnica de extracción de dos capas de influencia. Para aumentar la precisión de esta técnica se han utilizado zonas circulares en las cercanías de la fuente. Dichas zonas expresan de forma más certera la simetría de los campos provocada por la fuente. No obstante, todas las pruebas realizadas indican que sin una adecuada técnica de cancelación de la singularidad el análisis preciso de circuitos encapsulados no es posible.

El motivo por el cual se emplean redes neuronales en la aproximación de las funciones de Green ha quedado cumplido. En efecto, las RBFNNs entrenadas y usadas en la resolución de la IE mediante el MoM consiguen una gran aceleración en el análisis de circuitos prácticos. El llamado método de análisis neuronal es cientos de veces más rápido que la solución directa basada en la evaluación directa de las funciones de Green. Ahora bien, para conseguir la división del espacio de entrada y la totalidad de RBFNNs es necesaria la inversión de un tiempo inicial. Se ha podido comprobar que este tiempo es significativo ya que incluso con la técnica de eliminación más eficaz se cifra en decenas de horas. Para reducir este periodo de forma apreciable se han aplicado diversas simplificaciones, en algún caso novedosas, sobre el algoritmo de entrenamiento OLS. Las modificaciones realizadas no merman la precisión del método y son adecuadas para minimizar el tiempo de diseño neuronal con los tamaños máximos de neuronas utilizados. A pesar de que, incluso con las simplificaciones empleadas, el tiempo de diseño total es elevado, hay que tener en cuenta que este periodo es requerido una única vez antes del análisis circuital. Una vez entrenadas las RBFNNs para una estructura y ancho de banda determinados, no es necesario ningún re-entrenamiento. Por tanto, gracias a la elevada ganancia temporal, el tiempo de diseño se convierte en despreciable conforme aumenta el número de análisis. Si comparamos el método desarrollado con la solución directa utilizada tan solo se requieren tres análisis circuitales para lograr que el método sea rentable en términos temporales.

En conclusión, se puede afirmar que el método neuronal es eficiente a pesar del tiempo inicial requerido. Ahora bien, el método presenta limitaciones e inconvenientes. Las redes neuronales son entrenadas únicamente para un estructura multicapa y un ancho de banda. Cualquier cambio significativo en las dimensiones de la caja, características de las capas dieléctricas o frecuencia de trabajo

conllevará la aparición de errores en el análisis del circuito. En consecuencia, el método será rentable si se van a realizar múltiples análisis en una misma estructura y ancho de banda. Esta situación se dará fácilmente si se deben diseñar uno o más circuitos en dicha estructura. El análisis es muy veloz pero puede ser poco competitivo en comparación con otras técnicas. Por ejemplo, si la cavidad no es eléctricamente grande en comparación con las dimensiones del circuito el análisis mediante una formulación espectral puede ser eficiente. Existen programas comerciales como *ADS*® que han alcanzado una elevada eficacia en el análisis de circuitos encapsulados. La exactitud en el análisis del método neuronal es satisfactoria y el tiempo de diseño es un periodo inicial que se vuelve despreciable con el número de simulaciones. Por tanto, para que el método neuronal incremente su rendimiento es necesario aumentar la ganancia temporal. En el Capítulo siguiente se desarrolla un nuevo tipo de red neuronal cuyo rendimiento es superior al de las RBFNNs. Estas redes neuronales propician un análisis más veloz e igual de preciso.

Capítulo 4

Método neuronal de análisis de circuitos apantallados multicapa basado en Redes neuronales de Chebychev

4.1. Introducción

En el Capítulo anterior se han desarrollado diversas técnicas que permiten la aproximación de las funciones de Green mediante RBFNNs. La sustitución de las funciones de Green por los valores de salida de las RBFNNs posibilita una gran reducción en el tiempo de análisis circuital en estructuras multicapa insertadas en cavidades metálicas. La tarea de aproximación de las funciones de Green es complicada debido principalmente a la presencia de la singularidad en la fuente. Además, tanto el espacio de entrada como de salida son multidimensionales, de modo que la cantidad de datos necesaria para representar completamente las variaciones de las funciones objetivo es elevada. Estos problemas han sido superados gracias a una técnica eficiente de eliminación de la singularidad y a la aplicación de algoritmos de división del espacio de entrada. A pesar de que las técnicas implantadas permiten la aproximación eficiente y precisa de las funciones de Green, las RBFNNs resultantes presentan tamaños de varios cientos de neuronas. Existe un importante margen para disminuir el tamaño neuronal de forma que el análisis circuital sea acelerado todavía en mayor medida.

El objetivo de este Capítulo consiste en la generación de nuevas redes neuronales que posean al menos la misma capacidad de aproximación que las RBFNNs, con un número mucho menor de neuronas. Como ya se comentó en la Sección 3.3.3, el algoritmo OLS, aunque es eficaz en el entrenamiento de las RBFNNs, no es óptimo en la selección de las neuronas de dichas redes neuronales. En este Capítulo, el mencionado algoritmo se aplica de forma novedosa a una red neuronal para alcanzar un rendimiento óptimo. Esta red neuronal posee una estructura similar a la de una RBFNN. En cambio, sus funciones neuronales están formadas por polinomios de Chebychev. Tal y como se demostrará en el Capítulo, el muestreo del espacio de entrada en las raíces de ciertos polinomios de Chebychev permite alcanzar el deseado rendimiento óptimo en el algoritmo OLS. Se consigue así la minimización del número de neuronas y a la vez la minimización del tiempo de entrenamiento. Estos

dos hechos incrementan de forma directa la eficiencia total del método de análisis neuronal. Además, los polinomios de Chebychev, al igual que otros polinomios y funciones como las de tipo gaussiano, presentan características adecuadas para su utilización como aproximadores de datos multidimensionales.

Los polinomios de Chebychev han sido poco utilizados en la aproximación de funciones o datos en el contexto de problemas electromagnéticos. Podemos citar el artículo [Borji y Safavi-Naeini, 2004] en el que se emplean productos de polinomios de Chebychev para aproximar las funciones de Green de cavidades rectangulares. En el mencionado trabajo las muestras de las funciones de Green, necesarias para calcular los coeficientes que multiplican a los productos de polinomios de Chebychev, se evalúan mediante la eficiente técnica de Ewald. Este procedimiento parece similar al método propuesto, sin embargo, el enfoque es diferente ya que los coeficientes se calculan mediante la integración de los polinomios. En el presente Capítulo, todos los cálculos se realizan en el dominio discreto de modo que se evita la evaluación numérica de integrales que involucran a las muestras de las funciones de Green y a los polinomios de Chebychev. Además, en [Borji y Safavi-Naeini, 2004] se utilizan todos los posibles productos de polinomios. Por tanto, no se efectúa ninguna selección guiada que posibilite la reducción del tamaño de términos de la fórmula de aproximación empleada. En nuestro trabajo, el algoritmo OLS permite la minimización del número de neuronas, de modo que el análisis circuital será acelerado en gran medida.

La precisión de las Redes neuronales de Chebychev entrenadas de forma novedosa es similar o superior a la de las RBFNNs entrenadas con métodos supervisados como el OLS. Para lograr la mejor representación de las funciones de Green, las Redes neuronales de Chebychev utilizan las técnicas de eliminación de las singularidad del Capítulo anterior y los algoritmos de división del espacio de entrada. Los resultados obtenidos demuestran que las redes neuronales basadas en polinomios de Chebychev permiten que el método neuronal de análisis sea competitivo frente a otras técnicas de análisis veloces o programas comerciales de análisis electromagnético.

Una de las limitaciones más importantes del algoritmo OLS radica en la necesidad de almacenaje de los regresores para acelerar la selección de aquellos regresores más apropiados. Si el conjunto de entrenamiento sobrepasa un límite máximo, los regresores no se pueden guardar en la memoria RAM impidiendo la ejecución del algoritmo. Esta limitación se traduce de forma práctica en una restricción del número de dimensiones del espacio de entrada en el problema de aproximación. En nuestro caso, las dimensiones se han limitado a las cuatro coordenadas espaciales que definen los puntos fuente y observación y a la variable frecuencia. Esta elección restringe la aplicación del método neuronal a una cavidad de estructura fija. En las Redes neuronales de Chebychev, si el espacio de entrada es muestreado de forma adecuada, el algoritmo OLS no requiere del almacenamiento de ninguna matriz de regresores. Por lo tanto, no existe ninguna limitación en el tamaño del conjunto de entrenamiento. Al desaparecer las restricciones respecto al número de dimensiones se pueden incluir nuevas coordenadas que codifiquen la variación de las dimensiones de la cavidad o de las características del dieléctrico. Los resultados demuestran que es posible incorporar estas variaciones, de modo que se pueden analizar diferentes filtros en cavidades distintas. Sin embargo, el tiempo de entrenamiento de todas las redes neuronales necesarias alcanza tal duración que convierte en poco práctico el método neuronal para cavidades variables. La ampliación del método neuronal a espacios de entrada-salida más complejos constituye una de las tareas a mejorar en futuras investigaciones.

La estructura de este Capítulo se describe a continuación. En primer lugar se analiza la eficiencia del algoritmo OLS en las RBFNNs, (Sección 4.2). En las dos siguientes secciones se explica la solución propuesta que posibilita el rendimiento óptimo del algoritmo OLS. Así, en la Sección 4.3 se definen las Redes neuronales de Chebychev mientras que en la Sección 4.4 se describe la aplicación del algoritmo OLS en este tipo de redes. Dicha aplicación requiere de una adaptación a las Redes neuronales de Chebychev y de un muestreo apropiado del conjunto de entrenamiento. Sin este muestreo, fundamentado en las raíces de los polinomios de Chebychev, el algoritmo no consigue la máxima eficiencia. La Sección se subdivide en tres sub-secciones que explican el algoritmo para espacios de entrada y salida unidimensionales y multidimensionales. El Capítulo continúa con el estudio de la capacidad de aproximación de las Redes neuronales de Chebychev, (Sección 4.5). Se demostrará que estas redes neuronales, sin depender de uno u otro tipo de entrenamiento, son aproximadores universales al igual que las RBFNNs. De forma teórica, esta clase de redes neuronales es capaz de aproximar cualquier función en el intervalo de trabajo. Esta característica, unida a la aptitud del algoritmo OLS, proporciona a las Redes neuronales de Chebychev grandes capacidades en la aproximación de funciones.

En las dos siguientes secciones se comprueba este hecho. De esta manera, en la Sección 4.6 se analiza el número de operaciones requeridas para seleccionar los regresores, es decir, las neuronas de una Red neuronal de Chebychev. El análisis se realiza principalmente mediante la comparación de la carga computacional del algoritmo OLS en las Redes neuronales de Chebychev y las RBFNNs. Para todos los tamaños de entrenamiento y número de neuronas, el novedoso desarrollo del algoritmo OLS supera con creces a la implementación más veloz de la RBFNN. En la Sección 4.7, se ponen a prueba las capacidades de aproximación mediante exigentes tests numéricos. La comprobación de la habilidad de las Redes neuronales de Chebychev es completada con el estudio y análisis de la aproximación de las funciones de Green en la Sección 4.8. A continuación, en la Sección 4.9, se lleva a cabo un resumen de las ventajas y desventajas de las nuevas Redes neuronales de Chebychev frente a las RBFNNs. En esta Sección se explica que las desventajas más importantes de las Redes neuronales de Chebychev son evitadas en el problema de aproximación de las funciones de Green. Posteriormente, en la Sección 4.10, las Redes neuronales de Chebychev son utilizadas en la aproximación de las funciones de Green en diferentes cavidades. Los algoritmos de diseño y división del espacio de entrada empleados en el Capítulo anterior son aplicados de nuevo con éxito. Las redes neuronales obtenidas sustituyen completamente a las funciones de Green evaluadas numéricamente en la resolución de la IE. Así, en esta Sección se muestra la gran aceleración del análisis de circuitos encapsulados. Además, la precisión lograda es tan elevada como la conseguida con las RBFNNs. La eficiencia del método neuronal es comparada con la conseguida con otros métodos de análisis electromagnético y programas comerciales, (Sección 4.11). Se demuestra que el método neuronal es competitivo frente a otras técnicas y herramientas de análisis gracias a la velocidad en la obtención de las respuesta electromagnética. Además, el método neuronal es una formulación espacial, por lo que presenta ventajas frente a las formulaciones espectrales cuando la cavidad es grande eléctricamente respecto al circuito impreso. La Sección termina con un estudio del número de análisis circuitales que permiten que el método neuronal sea rentable. El método de entrenamiento OLS en las Redes neuronales de Chebychev carece de las limitaciones respecto al uso de memoria RAM que poseían las RBFNNs. Gracias a esta ventaja, las funciones de Green son aproximadas en cavidades en las que se permite la variación de ciertas características. En la Sección 4.12, se muestra el análisis circuital

mediante Redes neuronales de Chebychev que incorporan las variaciones de las funciones de Green respecto a la permitividad relativa y grosor del dieléctrico de la estructura microtira o respecto a la variación en las dimensiones transversales (a y b) de la caja. Por último, las conclusiones de este Capítulo son descritas en la Sección 4.13.

4.2. Eficiencia del algoritmo OLS en la selección de las neuronas de una Red neuronal de funciones de base radial

Como se explicó en la Sección 3.3.3, el algoritmo de entrenamiento OLS de una RBFNN selecciona de forma supervisada los centros de las neuronas. Cada centro produce un vector columna que denominamos regresor. El algoritmo selecciona en cada iteración aquel regresor que aporta el mayor incremento en la representación de la energía de la salida deseada. Desde otro punto de vista, se selecciona el regresor que maximiza la reducción del error en la representación de la energía de la salida deseada (ver ecuaciones (3.37) para el caso unidimensional y (3.55a) para el multidimensional). Para calcular la contribución de cada regresor a la energía de salida el algoritmo OLS lleva a cabo un proceso de ortogonalización sobre la matriz original de regresores \mathbf{P} . En cada paso del proceso de ortogonalización se añade a la base ortogonal aquel candidato que maximiza el parámetro: $\hat{g}_i = \frac{\mathbf{m}_i^T \mathbf{y}}{\mathbf{m}_i^T \mathbf{m}_i}$. Este valor indica el incremento en la expresión de la energía de la salida (ver ecuación (3.33)). En la anterior ecuación el vector \mathbf{m}_i es el resultado de ortogonalizar el regresor candidato \mathbf{p}_i respecto a los vectores que forman la base ortogonal. Gracias al procedimiento de ortogonalización, la reducción del error en la contribución de cada regresor a la energía de salida, o de forma equivalente la maximización de \hat{g}_i , se calcula de forma independiente. Al ortogonalizar los regresores candidatos, la contribución de cada regresor se encuentra desacoplada del resto de contribuciones. Los regresores ortogonales generan el mismo espacio vectorial $L(\mathbf{P}) = L(\mathbf{M})$ que los regresores originales de los que provienen [Golub y Loan, 1996]. Debido a que la salida de la RBFNN para el conjunto de entrenamiento es una combinación lineal de los regresores seleccionados, los valores de salida de la red neuronal para los regresores originales y para los regresores ortogonales será igual. De esta forma, los regresores originales y los regresores ortogonales alcanzan la misma precisión.

En [Sherstinsky y Picard, 1996] se demuestra que el algoritmo OLS no es capaz de generar la RBFNN más compacta en cualquier tarea de aproximación. Este hecho es debido a que originalmente los regresores \mathbf{p}_i no son ortogonales y por tanto las contribuciones de dichos regresores a la energía de salida se encuentran mezcladas. De este modo, aunque el método OLS calcula el regresor que permite el máximo alineamiento con la salida deseada en una iteración determinada, no es capaz de proveer la mejor representación de la salida en sentido global. Una serie de regresores seleccionados mediante este procedimiento no formará la RBFNN más pequeña ya que es muy probable que exista un conjunto de regresores que, de forma conjunta, representen de manera más precisa la salida objetivo. El problema del algoritmo OLS radica en que selecciona el regresor de forma individual, sin atender a la contribución que realiza de forma global con otros regresores. A pesar ser un algoritmo de entrenamiento basado en un proceso de ortogonalización, las neuronas formarán regresores que no son ortogonales. El proceso de ortogonalización sirve únicamente para calcular la aportación individual de cada regresor de forma independiente. Dicha independencia puede producir un exceso de neuronas ya que no se tienen en cuenta las propiedades globales de los distintos grupos de regre-

sores que se pueden elegir para componer la RBFNN. En conclusión, el máximo alineamiento con la salida en cada una de las iteraciones no implica que el alineamiento global producido por la RBFNN sea máximo.

El siguiente ejemplo, extraído de [Sherstinsky y Picard, 1996], muestra la incapacidad del algoritmo OLS para seleccionar el número mínimo de neuronas para un determinado nivel de error. El problema consiste en aproximar de forma eficiente el vector salida y con la matriz de regresores P de la ecuación (4.2):

$$y = \begin{pmatrix} -100 \\ 100 \\ 100 \end{pmatrix} \quad (4.1)$$

$$P = \begin{pmatrix} 1 & 0,606531 & 0,135335 \\ 0,606531 & 1 & 0,606531 \\ 0,135335 & 0,606531 & 1 \end{pmatrix} \quad (4.2)$$

La energía de la salida es igual a $y^T y = 30000$. El objetivo es conseguir una energía de salida de la RBFNN lo más cercana al valor anterior con el menor número de neuronas, es decir, de regresores. El método OLS selecciona en primer lugar el tercer regresor de la matriz P debido a que produce el máximo alineamiento con la salida. En la siguiente iteración se ortogonalizan las columnas primera y segunda de la matriz P , y se calcula el alineamiento de cada vector ortogonal con la salida. El primer regresor ortogonal proporciona el máximo alineamiento, de modo que es seleccionado en segundo lugar. Así, la matriz P de regresores ortogonales (también normalizados) es:

$$M_{OLS} = \begin{pmatrix} 0,1149 & 0,8973 & -0,4263 \\ 0,5152 & 0,3130 & 0,7979 \\ 0,8494 & -0,3113 & -0,4263 \end{pmatrix} \quad (4.3)$$

La contribución a la energía de salida de cada uno de los regresores es igual a $\hat{g}_1^2 = 15614,1$, $\hat{g}_2^2 = 8019,8$ y $\hat{g}_3^2 = 6366,2$. El algoritmo OLS efectúa una permutación sobre el orden de los regresores originales de modo que la nueva matriz P obtenida es:

$$P_{OLS} = \begin{pmatrix} 0,135335 & 1 & 0,606531 \\ 0,606531 & 0,606531 & 1 \\ 1 & 0,135335 & 0,606531 \end{pmatrix} \quad (4.4)$$

Si deseamos generar una RBFNN con una única neurona deberemos seleccionar la primera columna de la matriz (4.4). En cambio, si deseamos emplear dos regresores (neuronas), eliminaríamos la tercera columna de la anterior matriz. En este caso, el porcentaje de la energía explicada por los dos regresores elegidos es cercana al 80 %:

$$\frac{\hat{g}_1^2 + \hat{g}_2^2}{\mathbf{y}^T \mathbf{y}} \approx 0,79 \quad (4.5)$$

Sin embargo, los dos primeros regresores de la matriz original \mathbf{P} permiten una representación mucho más precisa de la salida. En efecto, si ortogonalizamos la matriz original sin alterar la posición de los regresores obtenemos:

$$\mathbf{M}_{opt} = \begin{pmatrix} 0,8494 & -0,4521 & 0,2724 \\ 0,5152 & 0,5976 & -0,6144 \\ 0,1149 & 0,6622 & 0,7405 \end{pmatrix} \quad (4.6)$$

Los regresores de esta matriz ortogonal poseen una contribución a la energía de salida de $\hat{g}_{opt1}^2 = 480,7$, $\hat{g}_{opt2}^2 = 29305,3$ y $\hat{g}_{opt3}^2 = 213,9$. Así, el porcentaje de contribución de los dos primeros regresores alcanza el 99 % de la energía de salida:

$$\frac{\hat{g}_1^2 + \hat{g}_2^2}{\mathbf{y}^T \mathbf{y}} \approx 0,99 \quad (4.7)$$

Esta precisión es debida a que el primer regresor es casi ortogonal respecto al vector de salida. Así, el segundo regresor, al ser ortogonalizado respecto al primero, produce un vector casi paralelo a la salida. El algoritmo OLS es incapaz de detectar esta situación ya que busca en cada iteración el mejor regresor. Por tanto, no selecciona en la iteración inicial el primer regresor sino el tercero, perdiendo por ello la oportunidad de representar fielmente la salida con tan solo dos regresores. La eficiencia del método depende del nivel de error que se desee conseguir en la aproximación. Si se desea representar como mínimo el 50 % de la energía de la salida el algoritmo OLS requiere únicamente el primer regresor seleccionado \mathbf{p}_3 . En cambio, la matriz original necesita los dos primeros regresores ya que si se emplea únicamente el primero se obtiene un porcentaje de expresión de la energía de salida de tan solo el 1,6 %. No obstante, si el conjunto de entrenamiento no es muy reducido y el nivel de error no es muy elevado, el algoritmo OLS no suministrará el número mínimo de neuronas.

Aunque en este ejemplo la función \hat{g}_i^2 disminuye en cada iteración, en general, la aportación a la energía de salida no es una función monótona. Una solución aparente consistiría en ordenar los regresores ortogonales generados por el algoritmo OLS y rechazar aquellos que ofrezcan los menores valores de \hat{g}_i^2 . No obstante, esta permutación sobre los regresores ortogonales impide recuperar los regresores originales. Cualquier permutación en las columnas de la matriz \mathbf{P} conduce a la obtención de una matriz \mathbf{M} diferente. De este modo, las energías de los regresores solo tienen sentido para un orden particular y concreto de los regresores originales. Para que los regresores posean una función de representación de la energía objetivo \hat{g}_i^2 que pueda ser ordenada de forma monótona decreciente es necesario que sean inicialmente ortogonales. Cuando esto ocurre, la proyección de cada vector sobre el vector de salida es independiente del resto de proyecciones, es decir, de contribuciones. De esta forma, la contribución energética de cada uno de los regresores se halla desacoplada del resto de aportaciones. En este caso, el conjunto total de regresores puede ser ordenado según la aportación de cada regresor a la expresión de la función de salida.

Por consiguiente, en una matriz \mathbf{P} de regresores ortogonales, para un cierto nivel de error de aproximación, el menor conjunto de regresores siempre es aquel compuesto por los M primeros regresores del conjunto ordenado que son capaces de superar el error fijado. Ninguna otra selección de los regresores de la matriz original podrá mejorar la precisión del anterior subconjunto sin emplear un número mayor de regresores. Asimismo, un subconjunto de regresores compuesto por M regresores siempre tendrá un error de aproximación superior al logrado por los M primeros regresores del conjunto ordenado por energías \hat{g}_i^2 decrecientes de forma monótona.

4.3. Solución al problema de selección de regresores mediante el algoritmo OLS: Redes neuronales de Chebychev

En general, las funciones de base radial empleadas en las neuronas de una RBFNN generan regresores no ortogonales. Así, los datos de entrenamiento producirán una matriz \mathbf{P} cuyas columnas o regresores no serán ortogonales. Como se ha comprobado en el apartado anterior, este hecho es responsable de la ineficiencia en el algoritmo OLS. La solución real a los problemas de detección del grupo de regresores que mejor aproximan los datos de salida consiste en partir de un conjunto de regresores de entrenamiento ortogonales. Por tanto, hay que encontrar una función o conjunto de funciones que al menos, al ser abastecido con los datos de entrenamiento, produzca una serie de regresores ortogonales. De esta forma, el método OLS podrá ser aplicado para obtener una serie de valores de energía monótonos decrecientes.

Existen diversas funciones que presentan ortogonalidad, tanto en el dominio continuo como en el dominio discreto. En nuestro caso estamos interesados en la ortogonalidad de tipo discreto debido a que las operaciones de entrenamiento de la red neuronal se realizan sobre un conjunto limitado de datos discretos. La propiedad de ortogonalidad discreta de la función debe cumplir ciertas condiciones para su empleo como función neuronal. En primer lugar, el rango de valores en los que se cumple la condición de ortogonalidad debe estar limitado. De este modo, el dominio original de la función a aproximar podrá ser trasladado al intervalo en el que las funciones neuronales son ortogonales. Por otra parte, las funciones ortogonales deben poseer buenas capacidades de aproximación al igual que las funciones gaussianas en las RBFNNs o las funciones de tipo logístico en los MLPs.

Los polinomios ortogonales son uno de los tipos de funciones ortogonales que proporcionan mejor precisión en la aproximación de funciones y conjuntos de datos. Por ejemplo, los polinomios de Hermite han sido empleados en el desarrollo de diversas redes neuronales de estructura similar a la de una RBFNN, ver [Ma y Khorasani, 2005, Zhao y Atkeson, 1996, Hwang et al., 1994]. Sin embargo, el rango de ortogonalidad discreta y continua en los polinomios de Hermite se extiende a lo largo de toda la recta real $(-\infty, +\infty)$. De este modo, el dominio de existencia de la función a aproximar no puede ser trasladado al rango de ortogonalidad. Los polinomios de Hermite de menor orden son ortogonales en un rango limitado de la recta real, por ejemplo los diez primeros polinomios de Hermite son ortogonales en el intervalo $(-5, 5)$. No obstante, esta solución solo permite la aproximación de un número limitado de funciones. En efecto, aquellos datos de salida que presenten rápidas variaciones serán aproximados de forma imprecisa ya que los mencionados polinomios presentan un comportamiento de variación lenta.

Los polinomios ortogonales de Legendre también han sido usados como funciones neuronales, ver [Tseng y Yang, 1995, Yang y Tseng, 1996]. En este tipo de polinomios, el rango de ortogonalidad se halla limitado al intervalo $[-1, 1]$. Sin embargo, únicamente presentan ortogonalidad de clase continua. Por lo tanto, el método OLS no puede ser aplicado de forma eficiente en las redes neuronales que hacen uso de estos polinomios. Los algoritmos de entrenamiento desarrollados en los trabajos citados se centran en la aplicación de métodos de optimización, basados en el uso del gradiente de la función error, para seleccionar los polinomios y pesos correspondientes. En nuestro problema, el tamaño mínimo de los diferentes conjuntos de entrenamiento se cifra en varios miles de muestras. Conforme aumenta el espacio de búsqueda del extremo global, aumenta la probabilidad de que el método de gradiente quede atrapado en un mínimo local. Así, los conjuntos de entrenamiento de los trabajos mencionados se limitan a pocos cientos de muestras (por ejemplo, 225 datos de entrenamiento en [Hwang et al., 1994]). Además, tanto los polinomios de Hermite como los polinomios de Legendre se calculan mediante fórmulas recursivas. Si el número de neuronas, y por tanto de polinomios, es elevado, entonces el tiempo de cálculo de la salida de las neuronas cuyos polinomios sean de orden superior será muy elevado. Aunque la red neuronal fuera muy precisa, el rendimiento sería muy bajo, debido a la lentitud en el cálculo de los valores de salida. Una posible solución consiste en guardar las expresiones de un número de polinomios, pero este procedimiento se vuelve poco práctico si el número de polinomios se cifra en varios cientos.

Los polinomios de Chebychev son un grupo de polinomios que presentan ortogonalidad continua y discreta en el intervalo $[-1, 1]$. En particular, los polinomios de Chebychev de primera especie poseen las siguientes expresiones según el orden del polinomio, [Press et al., 1997]:

$$\begin{aligned}
 T_0(x) &= 1 \\
 T_1(x) &= x \\
 T_2(x) &= 2x^2 - 1 \\
 T_3(x) &= 4x^3 - 3x \\
 T_4(x) &= 8x^4 - 8x^2 + 1 \\
 &\dots \\
 T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) \quad n \geq 1
 \end{aligned} \tag{4.8}$$

La última línea de la ecuación (4.8) es la fórmula recursiva que puede utilizarse para calcular cualquier polinomio de Chebychev de primera especie (en adelante los denominaremos polinomios de Chebychev). Una de las principales virtudes de este tipo de polinomios radica en que se pueden calcular sin recurrir a ecuaciones recursivas gracias a la siguiente fórmula trigonométrica:

$$T_n(x) = \cos(n \arccos(x)) \tag{4.9}$$

En la figura 4.1 se observan los cinco primeros polinomios de Chebychev. Se aprecia que todos los polinomios están limitados entre ± 1 de modo que los máximos son iguales a 1 y los mínimos a -1 .

El polinomio de Chebychev de orden N posee $N + 1$ extremos, máximos y mínimos, en el intervalo $(-1, 1)$ localizados en:

$$x_m = \cos\left(\frac{\pi m}{n}\right) \quad m = 0, 1, \dots, n \quad (4.10)$$

Como veremos más adelante ésta es una propiedad muy importante en la determinación de la capacidad de los polinomios de Chebychev en la aproximación de funciones. Las raíces de los polinomios de Chebychev también son calculadas de forma analítica. Así el polinomio de $T_n(x)$ tiene N ceros localizados en:

$$x_k = \cos\left(\frac{\pi(k - 1/2)}{N}\right) \quad k = 1, \dots, N \quad (4.11)$$

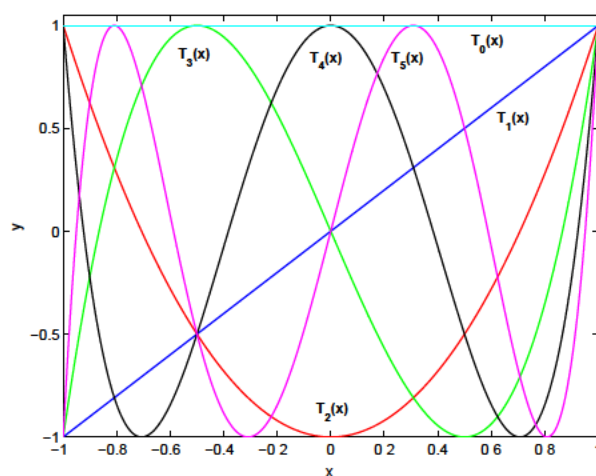


Figura 4.1: Los cinco primeros polinomios de Chebychev en el rango $[-1, 1]$.

En estos puntos se satisface la ortogonalidad discreta entre los polinomios de Chebychev. En efecto, los N primeros polinomios de Chebychev (desde $T_0(x)$ hasta $T_{N-1}(x)$) son ortogonales en los N ceros (x_k) del polinomio $T_N(x)$, [Press et al., 1997]:

$$\sum_{k=1}^N T_i(x_k) T_j(x_k) = \begin{cases} 0 & i \neq j \\ m/2 & i = j \neq 0 \\ m & i = j = 0 \end{cases} \quad (4.12)$$

La ortogonalidad discreta de los polinomios de Chebychev es utilizada para generar la bien conocida fórmula de aproximación de Chebychev [Press et al., 1997, Phillips, 2003]. Una función $f(x)$ es aproximada por la siguiente suma de polinomios:

$$F(x) \simeq \left[\sum_{j=1}^N c_j T_{j-1}(x) \right] - \frac{1}{2} \quad (4.13)$$

donde $x \in [-1, 1]$ y los coeficientes c_j se definen como:

$$\begin{aligned} c_j &= \frac{2}{N} \sum_{k=1}^N f(x_k) T_{j-1}(x) = \\ &= \frac{2}{N} \sum_{k=1}^N F \left[\cos \left(\frac{\pi(k-1/2)}{N} \right) \right] \cos \left(\frac{\pi(j-1)(k-1/2)}{N} \right) \end{aligned} \quad (4.14)$$

En la expresión (4.14) se utiliza la representación trigonométrica de los polinomios de Chebychev mostrada en (4.9). Los puntos x_k corresponden a las raíces del polinomio $T_n(x)$ de la ecuación (4.11). De este modo, los N primeros polinomios de la fórmula (4.13) son ortogonales, permitiendo una sencilla evaluación de los coeficientes c_j . Mediante un cambio de variable se puede aproximar una función que tenga límites arbitrarios $[x_{inf}, x_{sup}]$, ver [Press et al., 1997]:

$$x = \frac{x_{or} - \frac{1}{2}(x_{sup} + x_{inf})}{\frac{1}{2}(x_{sup} - x_{inf})} \quad (4.15)$$

donde x_{or} representa la variable independiente de la función objetivo definida en el rango original y x es el valor correspondiente en el rango de trabajo $[-1, 1]$.

Al igual que otros polinomios ortogonales, los polinomios de Chebychev han sido empleados en diversos tipos de redes neuronales. El primer trabajo que utilizó este tipo de polinomios en una red neuronal es [Namatame y Ueda, 1992]. Este trabajo es contemporáneo a la aparición del algoritmo de entrenamiento OLS para RBFNNs. Para entrenar las Redes neuronales de Chebychev utilizaba el bien conocido método de retro-propagación de los MLPs. Un enfoque diferente es aplicado en [Lee y Jeng, 1998]. En este artículo los polinomios de Chebychev aproximan a las funciones neuronales o funciones de activación de otros tipos de redes neuronales. Cada función neuronal se convierte en una suma de polinomios. De este modo, los polinomios de Chebychev se introducen en la red neuronal original sustituyendo las funciones a las que aproximan. El entrenamiento se limita al cálculo de un conjunto de pesos que agrupan los coeficientes de la aproximación de las funciones neuronales y los pesos de la red neuronal original. Gracias a la capacidad de representación de otros tipos de redes neuronales con capacidades de aproximación universal, el modelo propuesto en [Lee y Jeng, 1998] basado en polinomios de Chebychev tiene la habilidad de la aproximación universal. Este tipo de red neuronal fue posteriormente aplicado en diversos problemas prácticos [Weng et al., 2007, Purwar et al., 2007]. En [Chen y Tseng, 2004] se desarrolla una red neuronal con una estructura similar a la de una RBFNN. Así, la capa oculta de neuronas está compuesta por polinomios de Chebychev mientras que la capa de salida es lineal. En este caso, la propiedad de ortogonalidad expresada en (4.12) es empleada para calcular fácilmente los pesos de la capa lineal. El trabajo implementa dos métodos de entrenamiento diferentes. En el primero, los pesos de la red neuronal son obtenidos mediante descenso por gradiente. En el segundo, el entrenamiento se realiza con las muestras correspondientes a las raíces de Chebychev una vez realizado el cambio de variable descrito en (4.15). Si las N muestras originales no coinciden con las N raíces del polinomio $T_N(x)$, se aplica la técnica de aproximación de Lagrange para obtener dichas raíces a partir de las muestras disponibles.

Por ello, el procedimiento se denominó método numérico. El trabajo consiste en la adaptación de la fórmula de aproximación de Chebychev a una estructura neuronal. Las dos contribuciones originales de [Chen y Tseng, 2004] consisten en la obtención de las raíces a partir de los datos originales y la generación de polinomios ortogonales compuestos por polinomios de Chebychev en el caso de que la variable independiente x sea multidimensional.

Los trabajos anteriores se centran en el cálculo de los pesos que conforman ciertas capas de la red neuronal. Los polinomios de Chebychev son seleccionados como funciones neuronales por su capacidad en la aproximación de funciones. Sin embargo, ninguno de los trabajos propone un método de selección de neuronas, es decir, de elección de unos polinomios o combinaciones de polinomios en lugar de otros. Las propiedades expresadas en las ecuaciones (4.9) y (4.12) convierten a los polinomios de Chebychev en funciones neuronales aptas para el uso eficiente del algoritmo de entrenamiento OLS. En efecto, si durante la fase de entrenamiento la función objetivo se muestrea en los puntos correspondientes a los ceros del polinomio de orden N entonces los regresores formados con los N primeros polinomios de Chebychev (desde $T_0(x)$ hasta $T_{N-1}(x)$) serán ortogonales. De esta manera, el algoritmo OLS, especialmente el criterio de reducción del error en la aproximación de la función objetivo, podrá seleccionar de forma inequívoca aquellos regresores con mayor alienación con los datos de salida. En el presente Capítulo se propone la aplicación del algoritmo OLS en una red neuronal compuesta por polinomios de Chebychev. En la siguiente Sección, se explica la generación y entrenamiento de las Redes neuronales de Chebychev, tanto para el caso unidimensional como para el caso más general con espacios de entrada y salida multidimensionales.

4.4. Redes neuronales de Chebychev: algoritmo de entrenamiento

4.4.1. Espacio de entrada unidimensional-espacio de salida unidimensional

En esta Sub-sección se explica el algoritmo de entrenamiento para las Redes neuronales de Chebychev cuando realiza una transformación del tipo: $\mathfrak{R} \longrightarrow \mathfrak{R}$. El espacio de entrada y el espacio de salida son unidimensionales. La estructura de las Redes neuronales de Chebychev se muestra en la Figura 4.2. Al igual que las SVMs o RBFNNs existen tres capas diferentes. En la capa inicial simplemente se toman los datos de entrada. En la capa intermedia, el dato de entrada es introducido en cada neurona para producir un valor de salida definido su polinomio propio. En cada polinomio el subíndice $o(1)$ indica el orden del polinomio. Como deseamos obtener regresores ortogonales, en el caso de muestrear el espacio de entrada con las N raíces del polinomio $T_N(x)$, la red neuronal poseerá un tamaño máximo de N neuronas que obligatoriamente estarán formadas por los N primeros polinomios de Chebychev. Si se utiliza el tamaño máximo de la red neuronal, es decir, si se emplean todos los polinomios de Chebychev como en el trabajo [Chen y Tseng, 2004], para N datos de entrenamiento tendremos que $M = N$ con $o(1) = 0$, $o(2) = 1$, $o(3) = 2$ y así sucesivamente hasta la última neurona en la que $o(M) = N - 1$. En cambio, si seleccionamos aquellos polinomios que mejor se alineen con la salida, el orden del polinomio de la primera neurona no tiene por qué ser 0, así como el de la segunda neurona no tiene por qué ser 1, y del mismo modo para el resto de neuronas. En general, si se emplean N datos de entrenamiento el orden y se seleccionan M neuronas, la neurona j -ésima $o(j)$ será un valor comprendido entre 0 y $N - 1$. Una de las diferencias con la estructura de una RBFNN mostrada en (3.5) estriba en la ausencia de una neurona cuyo valor de salida sea constante.

Las Redes neuronales de Chebychev tendrá una neurona de esta clase si se selecciona el polinomio de orden cero, cuyo valor es igual a uno para cualquier x . Finalmente, en la capa de salida se realiza una transformación lineal para obtener el valor total de salida de la red neuronal.

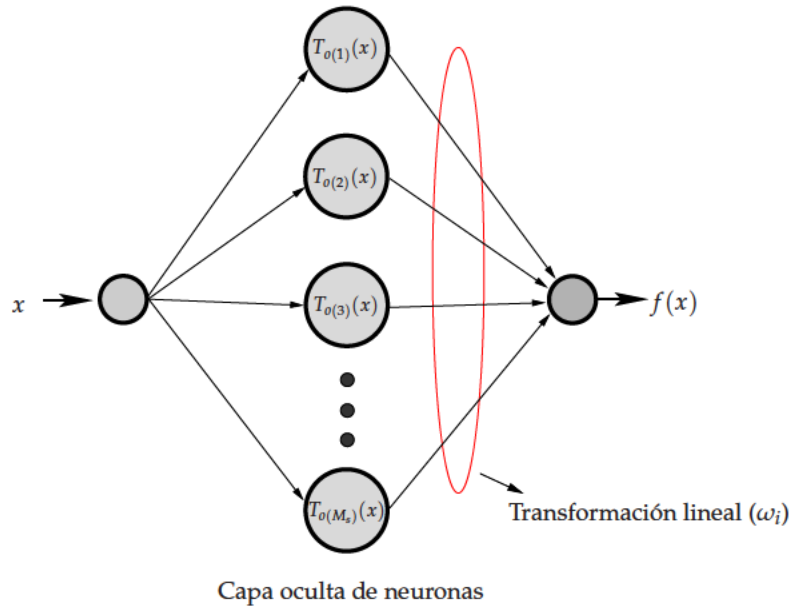


Figura 4.2: Estructura de una Red neuronal de Chebychev. Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. Cada neurona de la capa oculta realiza una transformación no lineal definida por un polinomio de Chebychev particular. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida.

Como se ha comentado en la parte final de la Sección 4.3 el algoritmo OLS expuesto en la Sección 3.3.3 se aplica en el entrenamiento de la nueva red neuronal. En primer lugar, la operación realizada por la red neuronal debe ser expresada en forma matricial:

$$\mathbf{y} = \mathbf{T}\boldsymbol{\omega} + \mathbf{E} \quad (4.16)$$

donde \mathbf{y} es el vector columna de muestras de salida, \mathbf{T} es la matriz de regresores, $\boldsymbol{\omega}$ es el vector columna de pesos de la capa final, y \mathbf{E} es un vector columna que contiene los errores cometidos por la red neuronal. En la ecuación (4.17) se desglosan los elementos de cada uno de los términos:

$$\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_N]^T \quad (4.17)$$

$$\mathbf{T} = [\mathbf{T}_1 \ \mathbf{T}_2 \ \cdots \ \mathbf{T}_{M_s}] \quad (4.18)$$

$$\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \cdots \ \omega_{M_s}]^T \quad (4.19)$$

$$\mathbf{E} = [\epsilon_1 \ \epsilon_2 \ \cdots \ \epsilon_N]^T \quad (4.20)$$

El número de datos de entrenamiento es N , de este modo el número máximo de regresores o polinomios será asimismo N . El orden de todo polinomio de la red neuronal deberá tener un or-

den menor o igual que $N - 1$ para que se cumpla la condición de ortogonalidad en los datos de entrenamiento. La selección de regresores se puede realizar sobre un subconjunto de $M_s < N$ regresores (polinomios) del conjunto total, ver (4.18). Bajo ninguna circunstancia M_s puede ser mayor que el número de datos de entrenamiento N ya que en ese caso se perdería la ortogonalidad en aquellos polinomios de orden mayor o igual a N . Por consiguiente, el método se aplicaría de forma sub-óptima al igual que en el caso de la RBFNN. En general $M_s = N$, es decir, la selección de los mejores regresores tendrá en cuenta a todos los polinomios de orden estrictamente menor que N . Por lo tanto, a partir de ahora se considerarán N regresores iniciales. Cada regresor i -ésimo es fruto de la evaluación de todos los datos de entrenamiento en el polinomio correspondiente de orden $o(i) \in 0, 1, \dots, N - 1$. Si se desea que los regresores formados durante el entrenamiento sean ortogonales de forma inicial, es necesario que los datos de entrenamiento sean las raíces (x_k) del polinomio de orden N . A este polinomio lo denominaremos polinomio de orden superior N . Por tanto, el regresor i -ésimo será:

$$\mathbf{T}_i = [T_{o(i)}(x_k(1)) \ T_{o(i)}(x_k(2)) \ \cdots \ T_{o(i)}(x_k(N))]^T \quad (4.21)$$

donde $1 \leq i \leq N$. Tal como se explica en la Sección 3.3.3, el algoritmo OLS selecciona en la primera iteración aquel regresor que mejor se alinea con la función objetivo. Para ello define una función de reducción del error (3.37) de modo que es mayor cuanto mayor es el alineamiento, es decir, cuanto mejor se expresa la energía de la función objetivo. En las siguientes iteraciones se seleccionan el resto de regresores. Para calcular de forma independiente la contribución de cada regresor del grupo de candidatos a la energía de salida se realiza un proceso de ortogonalización. Una vez seleccionado un regresor del grupo de candidatos se incorpora a la base ortogonal y se repite el mismo proceso de ortogonalización para determinar las contribuciones de los regresores restantes.

La aplicación del algoritmo OLS a las Redes neuronales de Chebychev es muy sencilla debido a la propiedad de ortogonalidad. Así, el producto de la matriz \mathbf{T} por su transpuesta \mathbf{T}^T es una matriz diagonal \mathbf{H} :

$$\mathbf{H} = \mathbf{T}^T \mathbf{T} = \begin{pmatrix} T_0^T T_0 & 0 & \cdots & 0 \\ 0 & T_1^T T_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{N-1}^T T_{N-1} \end{pmatrix} \quad (4.22)$$

Siguiendo el mismo procedimiento que en la Sección 3.3.3 podemos calcular la energía de la función de salida como:

$$\begin{aligned} \mathbf{y}^T \mathbf{y} &= [\mathbf{T}\boldsymbol{\omega}]^T [\mathbf{T}\boldsymbol{\omega}] + \mathbf{E}^T \mathbf{E} = \boldsymbol{\omega}^T \mathbf{H} \boldsymbol{\omega} = \\ &= \omega_1^2 \mathbf{T}_0^T \mathbf{T}_0 + \omega_2^2 \mathbf{T}_1^T \mathbf{T}_1 + \dots + \omega_N^2 \mathbf{T}_{N-1}^T \mathbf{T}_{N-1} + \mathbf{E}^T \mathbf{E} \end{aligned} \quad (4.23)$$

Al igual que en la ecuación (3.36) observamos que la energía de salida es la suma de las contribuciones de los regresores más la energía del error cometido. Por tanto, se puede calcular la contribución de cada regresor mediante el cálculo del ratio de reducción del error:

$$err_i = \frac{\omega_i^2 \mathbf{T}_{i-1}^T \mathbf{T}_{i-1}}{\mathbf{y}^T \mathbf{y}} \quad \text{Para } 1 \leq i \leq N \quad (4.24)$$

donde ω_i es el peso de la red neuronal correspondiente al polinomio de orden $i - 1$:

$$\omega_i = \frac{\mathbf{T}_{i-1}^T \mathbf{y}}{\mathbf{T}_{i-1}^T \mathbf{T}_{i-1}} \quad (4.25)$$

En las Redes neuronales de Chebychev, gracias a que los datos de entrenamiento son las raíces de un polinomio de Chebychev, los regresores son ortogonales. Por lo tanto, no es necesario aplicar sucesivos procesos de ortogonalización para elegir los mejores regresores. La contribución de cada regresor inicial es independiente de las aportaciones del resto de regresores. Consecuentemente, todos los valores de reducción del error de los regresores se calculan en el inicio del algoritmo. Todas las consideraciones realizadas en la Sección 4.2 sobre las ventajas de utilizar regresores ortogonales son aplicables al método de entrenamiento OLS en las Redes neuronales de Chebychev. Así, la ordenación monótona decreciente de los valores de reducción del error indica de forma inequívoca el grado de aportación, de forma decreciente, a la energía de salida. Para seleccionar M polinomios no se requiere la ejecución de M iteraciones del algoritmo, simplemente hay que escoger los M primeros polinomios con mayor reducción del error. Estos constituyen la Red neuronal de Chebychev de M neuronas que mejor aproxima la función objetivo en los datos de entrenamiento. Si se desea alcanzar mayor precisión se pueden añadir neuronas (hasta un máximo de N) sin necesidad de calcular nuevos valores de reducción del error para los candidatos. La aportación a la energía de salida está determinada por el valor de reducción del error computado en el inicio del algoritmo. Es sencillo comprobar que los pesos de la ecuación (4.25) son los coeficientes de la fórmula de aproximación de Chebychev (4.14). De este modo, podemos concluir que en una Red neuronal de Chebychev unidimensional la aplicación del algoritmo OLS consiste en la utilización del ratio de reducción del error en la fórmula de aproximación (4.13). La utilización de dicho ratio permite la correcta selección de las neuronas o polinomios.

La disminución del tiempo de entrenamiento es importante para hacer más eficiente el método neuronal de análisis circuital. Podemos aplicar un procedimiento similar al empleado en el entrenamiento OLS con las RBFNNs para minimizar el número de operaciones. Si sustituimos los valores de los pesos, (4.25), en la ecuación (4.24) obtenemos la siguiente expresión de la reducción del error:

$$err_i = \frac{(\mathbf{T}_{i-1}^T \mathbf{y})^2}{(\mathbf{T}_{i-1}^T \mathbf{T}_{i-1}) \mathbf{y}^T \mathbf{y}} \quad (4.26)$$

El resultado de multiplicar un regresor con su traspuesto se obtiene de forma analítica como se aprecia en (4.12). De esta manera, el número de operaciones requeridas para evaluar los ratios de reducción del error se minimiza en las siguientes expresiones:

$$err_i = \begin{cases} \frac{(\mathbf{T}_{i-1}^T \mathbf{y})^2}{(N/2) \mathbf{y}^T \mathbf{y}} & i-1 \neq 0 \\ \frac{(\mathbf{T}_{i-1}^T \mathbf{y})^2}{N \mathbf{y}^T \mathbf{y}} & i-1 = 0 \end{cases} \quad \text{Para } 1 \leq i \leq N \quad (4.27)$$

Esta fórmula es la más rápida en el cálculo de los coeficientes de reducción del error. Sin embargo, este procedimiento no proporciona de manera más veloz todos los parámetros de una Red neuronal de Chebychev. Si seleccionamos de este modo los regresores, los pesos de la capa lineal se deben calcular posteriormente. Este paso no es necesario si conservamos los pesos calculados con la ecuación (4.25) en lugar de introducirlos en la fórmula (4.24). Debido a que los regresores son originalmente ortogonales, los pesos no tienen por qué ser calculados de nuevo. Para elegir a un regresor se utiliza su peso correspondiente. Dicho peso puede ser conservado para su inclusión en la capa lineal de la red neuronal final.

4.4.2. Espacio de entrada multidimensional-espacio de salida unidimensional

En este caso consideramos que la Red neuronal de Chebychev realiza una transformación no lineal entre un espacio de dimensión n y un espacio de salida unidimensional: $\mathbb{R}^n \rightarrow \mathbb{R}$. Así, los datos de entrada son vectores de dimensión n : $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$. Cada coordenada pertenece al rango $[-1, 1]$ de forma que se cumple: $\mathbf{x} \in [-1, 1]^n$. Al igual que en el espacio unidimensional, si la variable independiente de la función original pertenece a un rango distinto de $[-1, 1]$ se lleva a cabo un cambio de variable para trasladar cada coordenada al intervalo de trabajo:

$$x_m = \frac{x_{m_{or}} - \frac{1}{2}(x_{m_{or \sup}} + x_{m_{or \inf}})}{\frac{1}{2}(x_{m_{or \sup}} - x_{m_{or \inf}})} \quad \text{Para } 1 \leq m \leq n \quad (4.28)$$

donde $x_{m_{or \inf}}$ y $x_{m_{or \sup}}$ son los límites inferior y superior originales de la coordenada m -ésima, $x_{m_{or}}$ es el valor de la coordenada en el rango original y x_m es el valor de la coordenada en el rango de los polinomios de Chebychev $[-1, 1]$.

Para conseguir un conjunto de regresores ortogonales en el espacio multidimensional \mathbb{R}^n debemos definir una función neuronal basada en polinomios de Chebychev. Además, se debe aplicar un muestreo adecuado en cada una de las coordenadas. En este trabajo se ha optado por escoger como función neuronal el producto de n polinomios de Chebychev:

$$P_i(\mathbf{x}) = T_{o_1(i)}(x_1) T_{o_2(i)}(x_2) \cdots T_{o_n(i)}(x_n) \quad (4.29)$$

Como se observa en la ecuación anterior el polinomio i -ésimo toma como variable independiente el valor de la coordenada i -ésima. Cada polinomio tiene un orden particular, cada neurona tiene una función neuronal única, es decir, un producto de polinomios diferente del resto de productos. De

esta forma se posibilita una gran variedad de formas en la función neuronal. Existirán funciones neuronales de rápida variación fruto de la multiplicación de polinomios de orden elevado. Por el contrario, el producto de polinomios de menor orden dará lugar a funciones neuronales de variación lenta. Así, tanto si la función objetivo posee una evolución lenta como una de tipo rápido, la red neuronal podrá expresar adecuadamente las variaciones de los datos de salida. Si la función objetivo presenta comportamientos tanto de tipo rápido como lento, la red neuronal podrá expresar de forma apropiada su evolución. Aquellas componentes de la función objetivo de mayor frecuencia serán expresadas por los productos de polinomios de mayor orden mientras que las componentes de baja frecuencia serán representadas por los productos de polinomios de menor orden. En la Figura 4.3 se muestran dos ejemplos de funciones neuronales para un espacio bidimensional. La función dibujada con línea continua es obtenida con dos polinomios de orden pequeño mientras que la otra función es el producto de dos polinomios de orden mayor. Se observa claramente que la primera función presenta variaciones muchos más lentas que la función neuronal fruto de la multiplicación de los polinomios de mayor orden. Además, la función neuronal (4.29) permite la aparición de una neurona i -ésima de salida constante igual a 1 si sus órdenes son $o_1(i) = o_2(i) = \dots = o_n(i) = 0$.

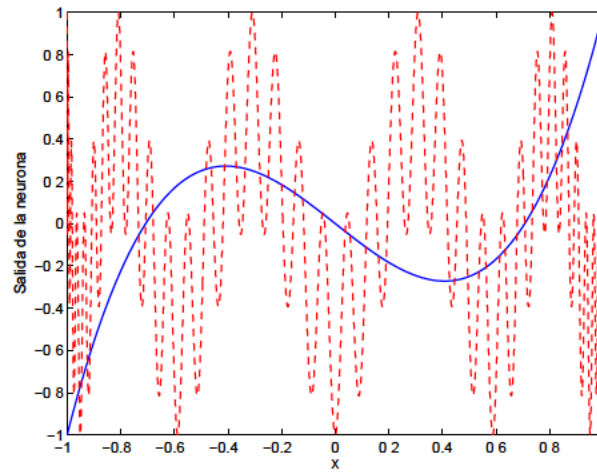


Figura 4.3: Función neuronal generada por la multiplicación del polinomio de Chebychev de primer y segundo orden (línea continua). Con guiones se dibuja la función neuronal obtenida con la multiplicación de los polinomios de orden 30 y 40.

Para completar la descripción de las Redes neuronales de Chebychev se requiere la definición de un muestreo de entrenamiento que posibilite la ortogonalidad de los regresores durante dicha fase. Para ello, la coordenada m -ésima será muestreada en las raíces del polinomio de orden superior $p(m)$. Si, por ejemplo, se desea que la coordenada m -ésima sea muestreada en diez puntos, entonces el polinomio de orden superior será $p(m) = 10$, los puntos de entrenamiento serán las raíces del polinomio de Chebychev de orden diez. La función neuronal tendrá diez posibles opciones en la coordenada m correspondientes a los polinomios con orden comprendido entre 0 y $p(m) - 1 = 9$. Este procedimiento ampliado a todas las coordenadas genera una rejilla regular o cubo multidimensional de puntos. En esta rejilla cada lado representa una dimensión o coordenada. Además, cada coordenada del espacio de entrada es muestreada con un número particular de raíces. Así, el polinomio de orden superior de la coordenada r , $p(r)$, puede ser distinto de $p(m)$ si $r \neq m$. El esquema en forma de rejilla produce un número de datos de entrenamiento igual a $N = p(1)p(2)\dots p(m)\dots p(n)$. Los puntos

de muestreo en la coordenada m -ésima son las raíces expresadas como:

$$x_{k_m(z)} = \cos \left(\frac{\pi(k_m(z) - 1/2)}{p(m)} \right) \quad (4.30)$$

donde $m = 1, 2, \dots, n$. El índice $k_m(z)$ pertenece al conjunto de valores $\{1, 2, \dots, p(m)\}$, donde z es un índice que varía desde 1 hasta el número total de datos N . Cada dato de entrenamiento se representa como un vector:

$$\mathbf{x}_{k(z)} = \{x_{k_1(z)}, x_{k_2(z)}, \dots, x_{k_n(z)}\} \quad (4.31)$$

Al igual que en las RBFNNs el número máximo de neuronas es igual al número de datos de entrenamiento. Asimismo, el muestreo efectuado para entrenar a la red neuronal impone una limitación en el orden de los polinomios de Chebychev que forman las funciones neuronales. Si la coordenada m -ésima es muestreada con $p(m)$ puntos, el polinomio correspondiente a dicha coordenada en la neurona i -ésima tendrá como orden máximo $p(m) - 1$ ($o_m(i) < p(m)$). Si evaluamos la función neuronal de la neurona i -ésima en un dato de entrada cualquiera (no de entrenamiento) de la red neuronal \mathbf{x} obtenemos:

$$P_i(\mathbf{x}) = T_{o_1(i)}(x_1) T_{o_2(i)}(x_2) \cdots T_{o_n(i)}(x_n) \quad \text{donde} \quad \begin{aligned} o_1(i) &\in \{0, 1, 2, \dots, p(1) - 1\} \\ o_2(i) &\in \{0, 1, 2, \dots, p(2) - 1\} \\ &\vdots \\ o_n(i) &\in \{0, 1, 2, \dots, p(n) - 1\} \end{aligned} \quad (4.32)$$

La limitación en el orden de los polinomios de cada coordenada está determinada obviamente por la propiedad de ortogonalidad mostrada en (4.12). La ortogonalidad de los regresores en el espacio de entrada multidimensional deberá cumplir la siguiente condición:

$$\mathbf{P}_i \mathbf{P}_j = \mathbf{P}_i(\mathbf{x}_k)^T \mathbf{P}_j(\mathbf{x}_k) = \sum_{z=1}^N P_i(\mathbf{x}_{k(z)}) P_j(\mathbf{x}_{k(z)}) \begin{cases} 0 & i \neq j \\ \neq 0 & i = j \end{cases} \quad (4.33)$$

donde $\mathbf{P}_i(\mathbf{x}_k)$ y $\mathbf{P}_j(\mathbf{x}_k)$ son los regresores fruto de evaluar respectivamente las funciones neuronales P_i y P_j en el conjunto de entrenamiento $\mathbf{x}_k = \mathbf{x}_{k(1)}, \mathbf{x}_{k(2)}, \dots, \mathbf{x}_{k(N)}$. La demostración de esta condición se explica en el Apéndice B.

El procedimiento de selección de neuronas descrito en la Sub-sección 4.4.1 para el caso unidimensional se aplica de la misma forma para espacios de entrada de dimensión mayor que uno gracias al cumplimiento de la condición descrita en (4.33). Como se ve en la ecuación (4.34) únicamente hay que sustituir los polinomios de Chebychev T_{i-1} por la función polinómica o regresor \mathbf{P}_i para obtener el ratio correspondiente. El ratio de reducción del error es utilizado de nuevo para detectar aquellos

regresores que presentan máxima alienación con la salida. Al igual que en el caso unidimensional el conjunto de M regresores así seleccionados ofrecerán la máxima precisión posible de entre todas las posibles combinaciones de M regresores utilizando los N regresores iniciales. Como siempre, la precisión va aparejada a un criterio de exactitud. En el caso del algoritmo OLS, a la máxima expresión de la energía de salida.

$$err_i = \frac{\omega_i^2 \mathbf{P}_i^T \mathbf{P}_i}{\mathbf{y}^T \mathbf{y}} \quad \text{Para } 1 \leq i \leq N \quad (4.34)$$

Los pesos de la capa de salida se calculan del mismo modo que en el caso unidimensional:

$$\omega_i = \frac{\mathbf{P}_i^T \mathbf{y}}{\mathbf{P}_i^T \mathbf{P}_i} \quad (4.35)$$

Dichos pesos pueden ser introducidos en la ecuación (4.34) para alcanzar una expresión que disminuye la carga computacional del método de elección de regresores:

$$err_i = \frac{(\mathbf{P}_i^T \mathbf{y})^2}{(\mathbf{P}_i^T \mathbf{P}_i) \mathbf{y}^T \mathbf{y}} \quad (4.36)$$

Como se comentó en el caso unidimensional, no es necesario aplicar esta fórmula ya que perdemos la oportunidad de conservar los pesos de la capa lineal. Si se emplea la ecuación (4.36), los pesos deben ser calculados una vez seleccionados todos los regresores perdiendo buena parte la eficiencia ganada.

4.4.3. Espacio de entrada multidimensional-espacio de salida multidimensional

En general, tanto el espacio de entrada como de salida serán multidimensionales: $\mathbb{R}^n \longrightarrow \mathbb{R}^q$. Ahora se trabaja con un vector de entrada \mathbf{x} n -dimensional y un vector de salida \mathbf{y} q -dimensional. La operación realizada por las Redes neuronales de Chebychev se expresa en forma matricial de igual forma que la mostrada en la ecuación (4.37a). La diferencia estriba en que los valores de salida agrupados en \mathbf{y} , los pesos ω y el error cometido \mathbf{E} son matrices de tamaño $N \times q$, donde N es el número total de datos de entrenamiento y por tanto de regresores. De nuevo, el objetivo es conseguir una expresión de la reducción del error en la aproximación de la función de salida. El desarrollo que se expone a continuación se ha realizado utilizando funciones neuronales genéricas nombradas como P_i . Cada función debe calcularse con la ecuación (4.29) si el espacio de entrada es multidimensional o con un polinomio de Chebychev aislado si el espacio de entrada es unidimensional. En la ecuación (4.37a) se desglosa cada elemento del sistema matricial:

$$\mathbf{y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_q] \quad (4.37a)$$

$$\mathbf{P} = [\mathbf{P}_1 \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_N] \quad (4.37b)$$

$$\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \cdots \ \omega_q] \quad (4.37c)$$

$$\mathbf{E} = [\epsilon_1 \ \epsilon_2 \ \cdots \ \epsilon_N] \quad (4.37d)$$

donde cada regresor \mathbf{P}_i es un vector columna de N elementos. Además, el vector columna de datos de salida y el vector columna de pesos j -ésimo se expresan como:

$$\mathbf{y}_\eta = [y_{1\eta} \ y_{2\eta} \ \cdots \ y_{N\eta}]^T \quad (4.38a)$$

$$\boldsymbol{\omega}_\eta = [\omega_{1\eta} \ \omega_{2\eta} \ \cdots \ \omega_{N\eta}]^T \quad (4.38b)$$

$$\text{para } \eta = 1, 2, \dots, q$$

Siguiendo la metodología empleada cuando el espacio de salida es unidimensional, debemos calcular en primer lugar la energía de salida:

$$\mathbf{y}^T \mathbf{y} = [\mathbf{P}\boldsymbol{\omega}]^T [\mathbf{P}\boldsymbol{\omega}] = \boldsymbol{\omega}^T \mathbf{H} \boldsymbol{\omega} + \mathbf{E}^T \mathbf{E} \quad (4.39)$$

donde \mathbf{H} es una matriz diagonal en la que el elemento no nulo de la posición (i, i) se calcula como: $\mathbf{P}_i^T \mathbf{P}_i$. Si el espacio de entrada es unidimensional entonces \mathbf{H} es la matriz expuesta en la ecuación (4.22). La energía de la señal de salida es la traza de la matriz producto $\mathbf{y}^T \mathbf{y}$. Por tanto debemos calcular la traza de la matriz $\boldsymbol{\omega}^T \mathbf{H} \boldsymbol{\omega}$ para obtener una expresión apropiada del coeficiente de reducción del error. Desarrollando los términos de la ecuación (4.39) obtenemos las siguientes matrices:

$$\begin{pmatrix} \mathbf{y}_1^T \mathbf{y}_1 & \square & \cdots & \square \\ \square & \mathbf{y}_2^T \mathbf{y}_2 & \cdots & \square \\ \vdots & \vdots & \ddots & \vdots \\ \square & \square & \cdots & \mathbf{y}_q^T \mathbf{y}_q \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N \omega_{i1}^2 (\mathbf{P}_i^T \mathbf{P}_i) & \square & \cdots & \square \\ \square & \sum_{i=1}^N \omega_{i2}^2 (\mathbf{P}_i^T \mathbf{P}_i) & \cdots & \square \\ \vdots & \vdots & \ddots & \vdots \\ \square & \square & \cdots & \sum_{i=1}^N \omega_{iq}^2 (\mathbf{P}_i^T \mathbf{P}_i) \end{pmatrix} + \mathbf{E}^T \mathbf{E} \quad (4.40)$$

Igualando ambas trazas obtenemos la expresión de la energía de salida en función de las contribuciones a dicha energía por parte de los regresores neuronales:

$$\text{Traza}(\mathbf{y}^T \mathbf{y}) = \sum_{\eta=1}^q \sum_{i=1}^N \omega_{i\eta}^2 (\mathbf{P}_i^T \mathbf{P}_i) + \mathbf{E}^T \mathbf{E} = \sum_{i=1}^N \mathbf{P}_i^T \mathbf{P}_i \sum_{\eta=1}^q \omega_{i\eta}^2 + \mathbf{E}^T \mathbf{E} \quad (4.41)$$

Finalmente, se consigue una expresión de la reducción del error en la representación de la energía de salida para el caso multidimensional:

$$err_i = \frac{\mathbf{P}_i^T \mathbf{P}_i \sum_{\eta=1}^q \omega_{i\eta}^2}{\text{Traza}(\mathbf{y}^T \mathbf{y})} \quad \text{Para } 1 \leq i \leq N \quad (4.42)$$

Se aprecia la similitud con la fórmula (3.55a) de la Sección 3.3.3 empleada en la versión del algoritmo OLS que se aplica en las RBFNNs. Al igual que en el caso unidimensional, los errores err_i de la ecuación (4.42) se calculan una sola vez en el inicio del procedimiento. De nuevo, la ordenación de forma monótona decreciente de los coeficientes de reducción del error permite la determinación de los mejores regresores, es decir, de las mejores funciones neuronales. Las mismas consideraciones sobre la capacidad de aproximación y de minimización del número de neuronas, expuestas en el final de la Sub-sección 4.4.1 para el caso unidimensional se aplican al caso multidimensional.

En esta Sección se ha mostrado la habilidad de los polinomios de Chebychev para formar una red neuronal capaz de aprovechar de forma óptima la capacidad de selección de regresores del algoritmo OLS. En la siguiente Sección se estudia la capacidad de aproximación de las redes neuronales así construidas tanto en el caso de un espacio de entrada unidimensional como en el caso multidimensional.

4.5. Capacidad de aproximación de las Redes neuronales de Chebychev

Las RBFNNs son aproximadores universales [Park y Sandberg, 1991]. Esta propiedad asegura que existe una RBFNN que aproxima de forma uniforme a toda función real continua dentro de un determinado intervalo de trabajo, cerrado y acotado, con el grado de precisión deseado. El algoritmo de diseño o entrenamiento intenta conseguir, con los datos disponibles, la RBFNN que aproxima con la máxima precisión la función objetivo. El rendimiento óptimo se alcanza cuando la red neuronal consigue la máxima precisión, con el menor número de neuronas. Como hemos visto en la Sección anterior, el algoritmo OLS es sub-óptimo en la selección de las neuronas en una RBFNN. Sin embargo, es capaz de elegir un conjunto de neuronas más reducido que otros algoritmos de entrenamiento como el de selección aleatoria o el de K-medias ("K-means"). El algoritmo OLS alcanza un resultado óptimo en las Redes neuronales de Chebychev tanto unidimensional como multidimensional, si ésta es entrenada con las raíces de los polinomios adecuados. Ahora bien, si la capacidad de aproximación de las Redes neuronales de Chebychev es mucho menor que otras redes neuronales como las RBFNNs, la ventaja generada por el algoritmo OLS desaparece. En efecto, aunque el algoritmo OLS seleccione el conjunto más compacto de neuronas en una Red neuronal de Chebychev para una determinada precisión, puede darse el caso de que la precisión máxima alcanzada sea muy inferior a la conseguida por una RBFNN. Otro resultado insatisfactorio, derivado de la menor capacidad de aproximación, consiste en la necesidad de un mayor tamaño neuronal que una RBFNN para realizar correctamente una tarea determinada. A pesar de que el algoritmo OLS sea óptimo en una red neuronal, hay que demostrar que la propia red neuronal posee la capacidad de aproximar todo tipo de funciones con gran precisión. Es decir, hay que demostrar que las Redes neuronales de Chebychev son aproximadores universales. Para ello, hemos recurrido a la aplicación del teorema de Stone-Weierstrass a las Redes neuronales de Chebychev. La enumeración y explicación de las condiciones de este teorema pueden consultarse en el Apéndice C. A continuación comentamos brevemente la

capacidad de aproximación de los polinomios de Chebychev y las consecuencias del cumplimiento del mencionado teorema para el caso de un espacio de entrada multidimensional.

En el caso de que el espacio de entrada sea unidimensional, las Redes neuronales de Chebychev equivalen a una suma de polinomios de Chebychev. Las habilidades de los polinomios en la aproximación de funciones han sido estudiadas ampliamente. El teorema original de Weierstrass sobre aproximación universal se refiere exclusivamente a polinomios. Cualquier función continua en un intervalo cerrado y acotado, puede ser aproximada de forma uniforme por polinomios con cualquier grado de precisión. Por tanto, la Red neuronal de Chebychev en el caso unidimensional constituye un aproximador universal. No obstante, en el Apéndice C se aplica el teorema de Stone-Weierstrass para demostrar efectivamente que en este caso las Redes neuronales de Chebychev son aproximadores universales.

La capacidad de los polinomios de Chebychev para aproximar funciones está además respaldada por su forma en el intervalo $[-1, 1]$. La forma de los polinomios de Chebychev es tal que cumplen la propiedad de equioscila. Si una función posee $n + 2$ máximos de igual valor absoluto, en los que los signos de dichos máximos son alternativos entonces se dice que la función equioscila en esos $n + 2$ puntos [Phillips, 2003]. El polinomio de Chebychev de orden $n + 1$ tiene $n + 2$ máximos que equioscilan. Así, el polinomio T_{n+1} es igual a $(-1)^{j-1}$ en sus extremos x_j para $j = 1, 2, \dots, n + 2$, ver (4.10). El polinomio aproximador p_n cuya función de error $|F - p_n|$ equioscila constituye el polinomio minimax de la función a aproximar. El polinomio minimax de un orden determinado, minimiza el máximo error de la aproximación. Por ello, es el polinomio de ese orden que mejor aproxima la función objetivo. Para un orden determinado n existe una única aproximación minimax para la función F . Si los polinomios de Chebychev equioscilan, es por tanto plausible pensar que pueden constituir el polinomio minimax de un determinado orden.

La fórmula de aproximación de Chebychev mostrada en (4.13) permite aproximar una función $f(x)$ mediante la suma de N polinomios de Chebychev. Como los polinomios de Chebychev son un aproximador universal podemos aumentar el número de polinomios N hasta que virtualmente la aproximación de la función $f(x)$ sea perfecta. Podemos truncar la fórmula (4.13) para obtener un sumatorio de polinomios de orden máximo m :

$$f(x) \approx \left[\sum_{k=1}^m c_k T_{k-1}(x) \right] - \frac{1}{2} c_1 \quad (4.43)$$

Este sumatorio forma un polinomio de orden m que constituye una aproximación cercana al polinomio minimax de $f(x)$. El valor de todos los polinomios de Chebychev se encuentra acotado entre -1 y 1 . De este modo, la diferencia entre la aproximación “exacta” del polinomio de orden N de la fórmula original y el polinomio truncado de orden m no puede ser mayor que el sumatorio de los coeficientes de los polinomios descartados $\sum_{k=m+1}^N c_k$ [Press et al., 1997]. En general los coeficientes decrecen rápidamente de forma que el error total respecto a la aproximación “exacta” estará dominada por el primer término $c_{m+1} T_m(x)$. Este término es una función que equioscila en $m + 1$ puntos, de modo que la función de error se considera una función que equioscila. La fórmula truncada de aproximación de Chebychev es un polinomio de fácil construcción, que resulta muy cercano

en rendimiento al polinomio minimax de orden $m - 1$ [Press et al., 1997].

El método de entrenamiento OLS empleado permite la selección de un conjunto de polinomios, de entre los N iniciales, que mejor aproximan la señal de salida. Esta serie de polinomios es aquella que proporciona la mayor reducción del error de aproximación. El valor de reducción del error (4.24) de un regresor es directamente proporcional al valor de su peso correspondiente. El peso es una medida de la proyección del regresor con el vector de salida. Por tanto, el peso indica el porcentaje de alineamiento con la salida deseada, y por ello es apropiado para calcular el error cometido por el regresor en la expresión de la energía de salida. En el algoritmo de entrenamiento OLS se ordenan los valores de reducción del error de forma monótona decreciente para seleccionar los mejores regresores. Consecuentemente, la ordenación es, en última instancia una ordenación del valor absoluto de los pesos ω_i .

Los pesos ω_i de la red neuronal coinciden con los coeficientes de la fórmula de aproximación de Chebychev. Así, el algoritmo de entrenamiento en el caso unidimensional consiste en una reordenación de términos de la fórmula (4.13). Al contrario que la fórmula truncada (4.43) la selección de m términos no tiene por qué implicar la selección de los m primeros términos de la serie total de polinomios. Al truncar la serie total de N términos se asume que los coeficientes c_k desde $m + 1$ hasta N toman valores decrecientes y de menor valor que los c'_k seleccionados. Sin embargo, este hecho no se puede asegurar. Se puede dar el caso de que el coeficiente c_{m+1} sea mayor que los coeficientes seleccionados debido a que la función de salida presenta rápidas variaciones que son expresadas correctamente por el polinomio correspondiente $T_m(x)$. Cuando se aplica el método de entrenamiento propuesto esta situación nunca puede aparecer. Una vez reordenados los términos, el nuevo término situado en la posición $m + 1$ tendrá un valor menor que el término de la posición m . La red neuronal así construida permite con un número igual de m términos un aproximación más precisa a la función deseada que la obtenida con fórmula truncada de la ecuación (4.43). En la red neuronal, el error está dominado por un término $\omega_{m+1} T_{o(m+1)}(x) \leq \omega_{m+1} \quad \forall x$ donde $o(m+1) \in 1, 2, \dots, N - 1$. Gracias al método de entrenamiento, ω_{m+1} será con seguridad menor que los pesos ya seleccionados (con la excepción del polinomio de orden cero, aunque lo habitual es que sea elegido entre las primeras neuronas).

El error total será igual a $\sum_{k=m+1}^N \omega_k$. Este término tiende a cero conforme m tiende a N . En la fórmula truncada los coeficientes no están ordenados y por tanto no es seguro que el error tienda con la misma velocidad a cero cuando m tiende a N . Una Red neuronal de Chebychev de m términos no constituye una aproximación mejor que la fórmula truncada al polinomio minimax de orden $m - 1$ ya que el polinomio de la posición j -ésima no tiene por qué poseer un orden igual a $j - 1$. Por tanto, la neurona m -ésima puede estar formada por un polinomio de orden mayor que $m - 1$. No obstante, el método de entrenamiento proporciona una red neuronal que aproxima con más precisión la función de salida que la fórmula truncada, con el mismo número de términos. Así, para un nivel de aproximación determinado la red neuronal será más compacta que la fórmula truncada de aproximación.

Para demostrar que las Redes neuronales de Chebychev constituyen aproximadores universales en el caso de un espacio de entrada multidimensional recurrimos de nuevo al teorema de Stone-Weierstrass. En el Apéndice C se demuestra que las Redes neuronales de Chebychev cumplen este

teorema para cualquier dimensión del espacio de entrada. Por lo tanto, dentro del espacio de trabajo multidimensional $[-1, 1]^n$, las Redes neuronales de Chebychev podrán aproximar cualquier función continua con el grado de precisión deseado.

4.6. Carga computacional en la selección de los regresores

La selección de los regresores, es decir, de las neuronas, mediante el algoritmo OLS es una tarea sencilla si dichos regresores son originalmente ortogonales. En este caso, únicamente hay que calcular las contribuciones de los regresores a la energía de salida para ordenarlas a continuación de forma monótona decreciente. Por lo tanto, el número de operaciones necesarias para diseñar la red neuronal será mucho menor que el requerido en la aplicación del algoritmo OLS en una RBFNN. En esta red neuronal los regresores no son ortogonales y en cada iteración es obligado ortogonalizar los regresores no seleccionados para calcular su contribución.

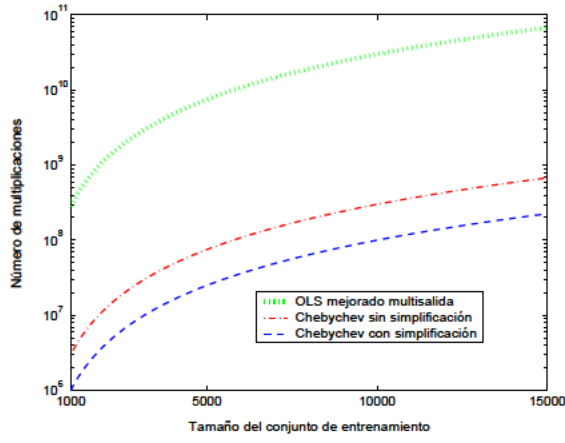
En esta Sección vamos a comparar la carga computacional del método de entrenamiento OLS en las RBFNNs y en las Redes neuronales de Chebychev. Al igual que en la Sección 3.9.1 se recurre al número de multiplicaciones realizadas en la selección de los regresores para estimar la carga computacional de cada método de entrenamiento. En la siguiente ecuación se muestra el número de multiplicaciones si se emplea la fórmula (4.34) para calcular el ratio de reducción del error en la expresión de la energía de salida:

$$\text{Numero de multiplicaciones sin simplificación} = Nq + N[(N + 2 + q(2N + 2))] \quad (4.44)$$

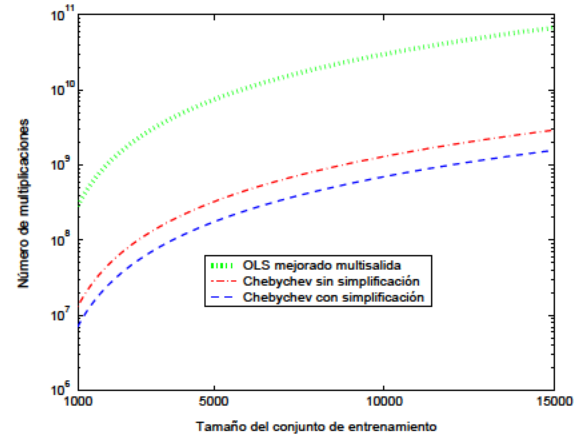
donde N es el número de datos de entrenamiento y q la dimensión del espacio de salida. Si se emplea la fórmula simplificada (4.36), el número de multiplicaciones es:

$$\text{Numero de multiplicaciones con simplificación} = Nq + N[q(N + 1) + (N + 1) + 1] \quad (4.45)$$

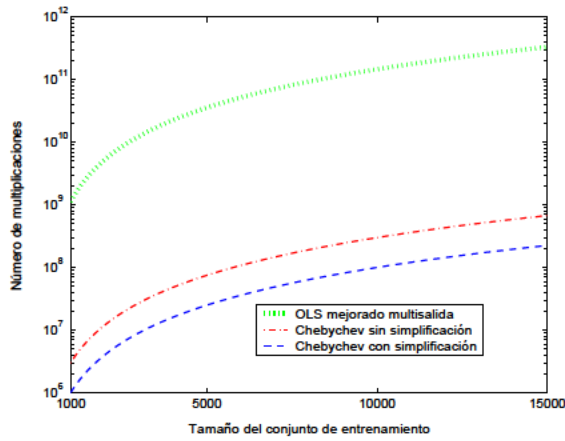
El número de multiplicaciones mínimo en el algoritmo OLS en una RBFNN para los tamaños de entrenamiento utilizados en la aproximación de las funciones de Green se alcanzaba con la versión desarrollada en la Sección 3.9.1. Ya sea con simplificación o sin simplificación, el algoritmo OLS en las Redes neuronales de Chebychev requiere un número menor de multiplicaciones que la versión utilizada en las RBFNNs, ver ecuación (3.67). En la Figura 4.4 se muestra el número de multiplicaciones requerido en el algoritmo OLS mejorado en las RBFNNs, en el algoritmo OLS en las Redes neuronales de Chebychev sin simplificación y en el algoritmo OLS en las Redes neuronales de Chebychev con simplificación para distintos tamaños de entrenamiento y un número fijo de neuronas. Así, las Figuras 4.4(a) y 4.4(b) muestran los resultados para un número reducido de 100 neuronas. Se aprecia que tanto en el caso unidimensional, como en el caso multidimensional, el número de multiplicaciones es hasta dos órdenes de magnitud menor en el caso de las Redes neuronales de Chebychev. Esta diferencia en la carga computacional se mantiene si el número de regresores seleccionados aumenta



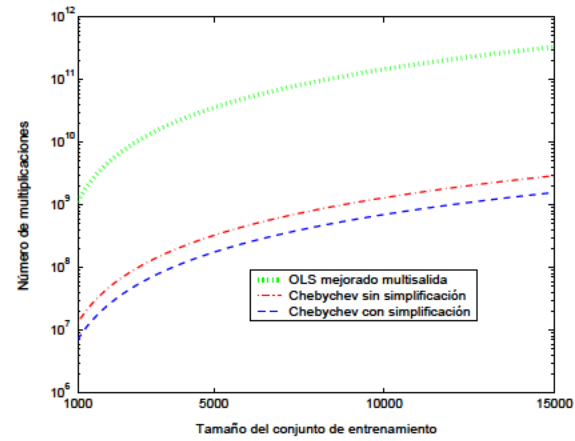
(a) Número de neuronas igual a 100. Salida unidimensional.



(b) Número de neuronas igual a 100. Dimensión del espacio de salida igual a 6.



(c) Número de neuronas igual a 500. Salida unidimensional.



(d) Número de neuronas igual a 500. Dimensión del espacio de salida igual a 6.

Figura 4.4: Número de multiplicaciones para los algoritmos FOLS en una RBFNN (línea punteada), OLS con simplificación (línea con guiones) y sin simplificación (línea con punto y raya) en las Redes neuronales de Chebychev.

hasta 500, Figuras 4.4(c) y 4.4(d). En todos los casos, la versión simplificada necesita menos multiplicaciones que la versión sin simplificación. Sin embargo, como ya se comentó en la Sub-sección 4.4.2, la versión simplificada requiere del cálculo de los pesos de la capa de salida. En cambio, en la versión sin simplificación los pesos se calculan durante la propia selección de los regresores. Por tanto, la diferencia en la carga computacional final entre ambas versiones no es apreciable. Además, como se comprobará en la Sub-sección 4.8.2, el tiempo empleado en el cálculo de los propios regresores a partir de los datos de entrenamiento es superior al tiempo de selección de los regresores. De este modo, ambas versiones arrojan resultados temporales parecidos sin que exista una ventaja significativa en ninguno de ellos.

4.7. Tests numéricos de la capacidad de aproximación de las Redes neuronales de Chebychev

En la anterior Sección se ha mostrado que las Redes neuronales de Chebychev tanto en el caso unidimensional como en el multidimensional constituyen aproximadores universales. Ahora bien, la propiedad de aproximación universal solo indica la existencia de una función generada por la Red neuronal de Chebychev que aproxima, con la precisión deseada, la función objetivo. Por ello, esta propiedad no explica cómo seleccionar los parámetros de la red neuronal. En las Redes neuronales de Chebychev empleamos el algoritmo OLS y su criterio de explicación de la energía de salida para conseguir un número mínimo de neuronas. Las RBFNNs también son aproximadores universales. Como ya se ha demostrado el algoritmo OLS no alcanza un rendimiento óptimo en las RBFNNs. De este modo, es previsible que las Redes neuronales de Chebychev consigan un rendimiento mayor en la aproximación de funciones que las RBFNNs entrenadas con el algoritmo OLS. Sin embargo, es muy adecuado realizar una serie de tests numéricos con diferentes funciones para comprobar que las Redes neuronales de Chebychev son más efectivas que las RBFNNs. Aunque ambas redes neuronales son aproximadores universales hay que tener en cuenta que las funciones de las neuronas son diferentes. En el caso de las RBFNNs tenemos funciones gaussianas mientras que la nueva red neuronal utiliza polinomios de Chebychev. Si las funciones gaussianas ofrecen una aproximación más precisa que los polinomios de Chebychev, entonces la Red neuronal de Chebychev podría llegar a requerir más neuronas que una red neuronal construida con gaussianas como la RBFNN. Este hecho podría aparecer incluso si se utiliza de forma óptima el algoritmo OLS, tal como ocurre en las Redes neuronales de Chebychev al muestrear en las raíces de los polinomios.

Los tests numéricos realizados emplean funciones con espacios de entrada unidimensionales y multidimensionales. Por el contrario, el espacio de salida es unidimensional en todos los casos. Se ha buscado cierta variedad en las características de las funciones y en los rangos de trabajo. De esta forma, se comprueba también que los cambios de variable mostrados en (4.15) y (4.28) son válidos para trasladar la función original al rango de trabajo de las Redes neuronales de Chebychev. Además, todas las funciones objetivo han sido utilizadas ampliamente en la literatura científica para probar la capacidad de aproximación de diversas redes neuronales.

En cada test numérico se compara el rendimiento de la nueva Red neuronal de Chebychev entrenada con el algoritmo OLS y una RBFNN entrenada asimismo con el algoritmo OLS. En la RBFNN se muestra, en cada función, el mejor resultado entre todos los obtenidos con un conjunto de valores de varianza suficientemente amplio (entre 0,010 y 1,5). La comparación se completa con una Red neuronal de Chebychev en la que la elección de las neuronas no está regida por el criterio de expresión de la energía de salida, es decir, sin estar entrenada con el algoritmo OLS. En el caso unidimensional esta red coincide con la fórmula de aproximación de Chebychev, ver (4.13). Los polinomios se eligen así de forma consecutiva. En el caso multidimensional, se eligen como m neuronas aquellas que generan los m primeros regresores o columnas de la matriz T . La red neuronal dependerá así del orden en el que se generen los regresores. Al igual que en el caso unidimensional se comienza por aquellas neuronas con productos de polinomios de menor orden. El orden seguido es semejante al expuesto en las Tablas B.1 y B.2, correspondientes a la generación de los datos de entrenamiento. De este modo, la primera neurona poseerá como función propia un producto de polinomios de orden cero, la

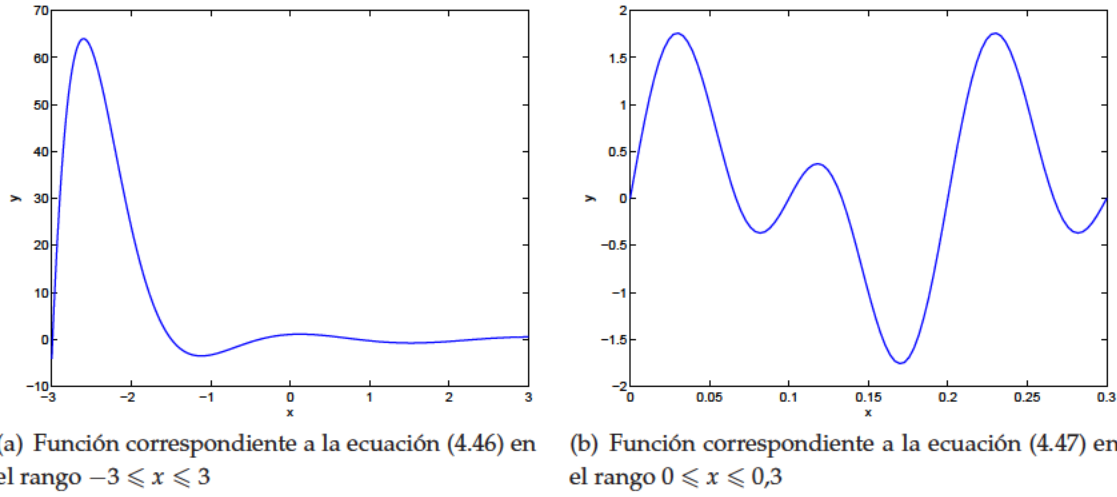


Figura 4.5: Funciones utilizadas en el test numérico de aproximación de funciones unidimensionales.

segunda un producto de polinomios de orden cero excepto en el polinomio de la última coordenada que poseerá orden igual a 1 y así sucesivamente. Nombraremos a la red neuronal así construida como Red neuronal de Chebyshev sin algoritmo OLS.

En primer lugar, se pusieron a prueba la capacidad de aproximación de las diferentes redes neuronales en funciones unidimensionales. La primera función utilizada en el test numérico corresponde con la función numerada como ejemplo 1 en el trabajo [Ma y Khorasani, 2005]. Su expresión se expone en la ecuación siguiente, (4.46):

$$f(x) = 0,2 \left[1 + \frac{1}{10}(x-7)^2 \right] \cos(2x) + 0,5e^{-2x} \sin(2x - 0,1\pi) \quad (4.46)$$

En [Ma y Khorasani, 2005], el rango de variación de x se sitúa entre 0 y 1. En nuestras simulaciones hemos optado por aumentar los límites del rango de variación de x para incrementar la dificultad del test. Así, la variable independiente toma valores en el conjunto $[-3, 3]$. De este modo se consigue una función de rápido crecimiento en una parte del dominio mientras que adopta una variación lenta en el resto del dominio, según se aprecia en la Figura 4.5(a).

Cada una de las tres redes neuronales fue entrenada con 100 puntos. En la Red neuronal de Chebyshev estos puntos corresponden a las raíces del polinomio de orden superior mientras que en la RBFNN los puntos de entrenamiento se generan de modo uniforme. Para validar cada red entrenada se generaron 200 puntos de forma uniforme en el dominio de x . En la Figura 4.6 se exhibe el error NMSE correspondiente a las tres redes neuronales para distintos tamaños neuronales. En este caso, a pesar del carácter sub-óptimo del algoritmo OLS en el entrenamiento de una RBFNN, ésta red neuronal consigue los mejores resultados cuando el número de neuronas es reducido. Sin embargo, las Redes neuronales de Chebyshev alcanzan menores niveles de error que la RBFNN conforme crece el número de neuronas.

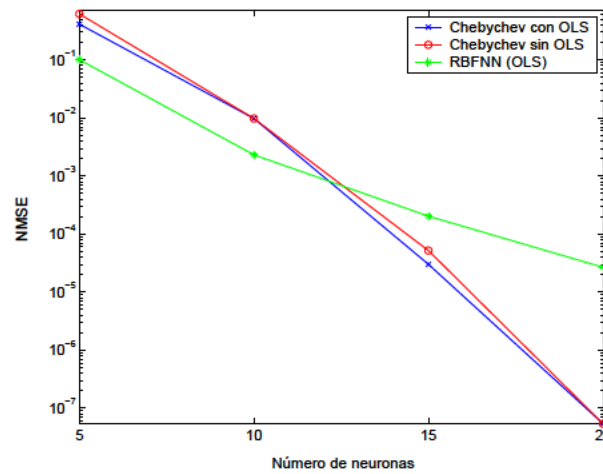


Figura 4.6: Error de tipo NMSE para las tres redes neuronales en la aproximación de la función de la ecuación (4.46).

La comparación entre los resultados de la nueva Red neuronal de Chebychev y la creada con la fórmula tradicional de aproximación muestra la capacidad de selección del algoritmo OLS. Con cinco neuronas la nueva Red neuronal de Chebychev consigue un error NMSE de validación de 0,4146, en cambio la fórmula de aproximación de Chebychev permite un NMSE de 0,6216. Cuando el tamaño de la red se limita a cinco neuronas, la fórmula tradicional constituye una red neuronal con los cinco primeros polinomios ($0 \leq \text{orden} \leq 4$). Por el contrario, el algoritmo OLS selecciona polinomios con orden mayor que 4 de forma que en este caso se incorpora en las cinco primeras neuronas un polinomio de orden 5 y otro de orden 6. Estos polinomios permiten reducciones del error en la expresión de la energía de salida mayores que las encontradas con los polinomios de orden 2 y 3, como se observa en la Figura 4.7. Con siete neuronas, la nueva red sigue siendo más eficiente ya que consigue un error NMSE de 0,0906 frente a un nivel de NMSE de 0,2604 por parte de la fórmula tradicional. A partir de un tamaño neuronal de 10 neuronas ambas redes presentan resultados similares debido a que seleccionan los mismos polinomios. Es decir, a partir de la décima neurona el algoritmo OLS selecciona polinomios con orden mayor que 9 y menor que 30 de forma prácticamente consecutiva. Este hecho es debido a que los ratios de reducción del error decrecen de forma monótona de forma natural, sin necesidad de reordenación, tal como se aprecia en la Figura 4.7.

La siguiente función es empleada en [Lee y Jeng, 1998] (ejemplo 4) para comprobar el rendimiento de un tipo de red neuronal basado en polinomios de Chebychev. En la Figura 4.5(b) se aprecia la evolución de la función cuyo comportamiento está determinado por la ecuación:

$$f(x) = \sin(2\pi 5x) + \sin(2\pi 10x) \quad (4.47)$$

En este caso hemos recurrido al estudio del error cuadrático medio para comparar el rendimiento de las tres redes neuronales. De este modo se facilita la comparación con la red neuronal desarrollada en [Lee y Jeng, 1998]. Al igual que en [Lee y Jeng, 1998], se utilizan 31 muestras de entrenamiento y 100 puntos de validación. El esquema de generación de los datos de entrenamiento y validación es

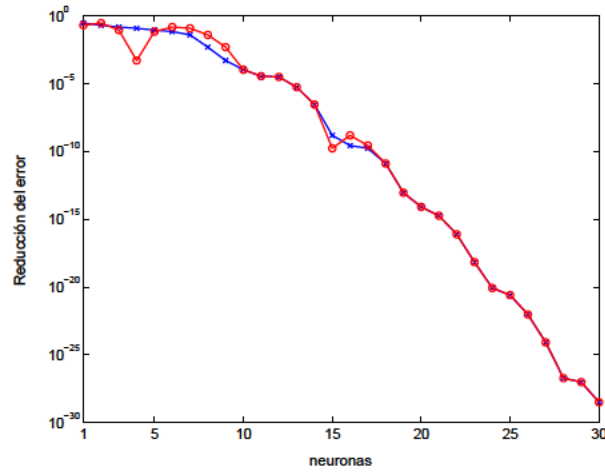


Figura 4.7: Ratio de reducción del error en la aproximación de la función de la ecuación (4.46). Los ratios ordenados de mayor a menor correspondientes a la aplicación del algoritmo OLS se dibujan con cruces. Los ratios desordenados, correspondientes a polinomios de orden creciente (sin algoritmo OLS), se dibujan con círculos.

el mismo que el empleado en la primera función. En las Tablas 4.1, 4.2 y 4.3 se exponen los valores de error de la nueva red neuronal, la fórmula tradicional (selección consecutiva de polinomios, sin entrenamiento OLS) y la RBFNN respectivamente. En este ejemplo, la nueva Red neuronal de Chebychev obtiene los mejores resultados prácticamente en todos los tamaños neuronales probados. En este caso, la nueva red neuronal conseguirá, bien redes más compactas y por tanto más rápidas, o bien redes más precisas. Tanto con un número reducido de neuronas como con un número elevado, la nueva red neuronal supera la efectividad de la RBFNN. Además, la RBFNN es incapaz de reducir el error cuando el tamaño neuronal crece. La nueva red neuronal no presenta este problema y sigue mejorando la precisión al aumentar el número de neuronas.

La ventaja del uso del algoritmo OLS con regresores ortogonales se muestra claramente en la Figura 4.8. La ordenación de los ratios de reducción del error permite generar una serie de términos que aumentan la precisión de la aproximación de forma creciente. En cambio, los ratios de reducción originales presentan una evolución fuertemente oscilante. De esta manera, si los polinomios son seleccionados de forma consecutiva, sin atender a su capacidad de aproximación, se seleccionarán polinomios de buena capacidad con otros que en poco mejorarán la precisión. La eficiencia de la aproximación será baja en este caso. En las Tablas 4.1 y 4.2 se observa que el error cuadrático medio de la fórmula tradicional es hasta tres órdenes de magnitud mayor que el alcanzado con la nueva red neuronal. La red neuronal desarrollada en [Lee y Jeng, 1998] obtiene en este ejemplo un rendimiento peor que la nueva red ya que alcanza un error cuadrático medio de 10^{-6} con 30 neuronas.

A continuación se muestran los resultados obtenidos en la aproximación de funciones cuyo espacio de entrada es multidimensional. Se han utilizado cuatro funciones con espacio de entrada bidimensional, previamente empleadas en la literatura científica en [Hwang et al., 1994], [Kwok y Yeung, 1997] y [Ma y Khorasani, 2005]. La comparación se completa con una quinta función cuyo espacio de entrada posee tres dimensiones, [Ma y Khorasani, 2005] (ejemplo 3). Las expresiones de cada una de las funciones se muestran en las ecuaciones (4.48). La forma de las cuatro primeras

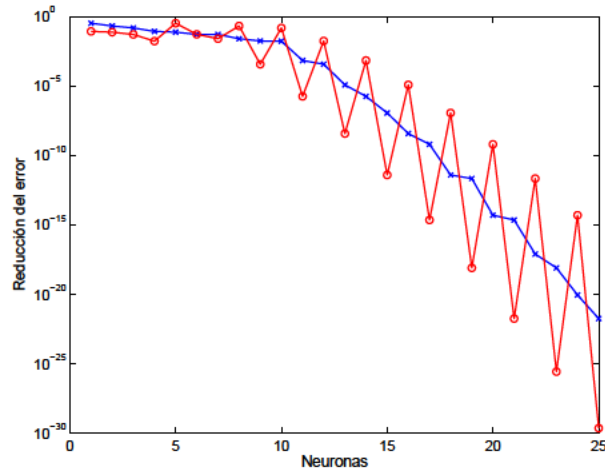


Figura 4.8: Ratio de reducción del error en la aproximación de la función de la ecuación (4.47).

Tabla 4.1: Error cuadrático medio del conjunto de entrenamiento y validación para la Red neuronal de Chebychev en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].

Neuronas	Entrenamiento	Validación
5	4.23	15.02
10	0.0267	0.0967
15	$1,01 \cdot 10^{-7}$	$3,83 \cdot 10^{-7}$
20	$5,79 \cdot 10^{-14}$	$2,01 \cdot 10^{-13}$
25	$2,02 \cdot 10^{-21}$	$6,64 \cdot 10^{-23}$

funciones se dibuja en las Figuras 4.9. El rango de variación en cada una de las dimensiones del espacio de entrada de todas las funciones se encuentra entre 0 y 1.

Tabla 4.2: Error cuadrático medio del conjunto de entrenamiento y validación para la Red neuronal de Chebychev sin algoritmo OLS en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].

Neuronas	Entrenamiento	Validación
5	11.79	45.26
10	0.4677	1.7144
15	$3,20 \cdot 10^{-4}$	$1,10 \cdot 10^{-3}$
20	$5,99 \cdot 10^{-11}$	$1,92 \cdot 10^{-10}$
25	$2,08 \cdot 10^{-16}$	$6,90 \cdot 10^{-16}$

Tabla 4.3: Error cuadrático medio del conjunto de entrenamiento y validación para la RBFNN en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].

Neuronas	Entrenamiento	Validación
5	12.99	40.88
10	0.0308	0.100
15	$5,40 \cdot 10^{-8}$	$2,86 \cdot 10^{-7}$
20	$1,48 \cdot 10^{-11}$	$1,55 \cdot 10^{-10}$
25	$1,48 \cdot 10^{-11}$	$1,55 \cdot 10^{-10}$

$$\text{Función AF: } f(x_1, x_2) = 1,3356 \left\{ 1,5(1 - x_1) + e^{2x_1-1} \sin(3\pi(x_1 - 0,6)^2) + e^{3(x_2-0,5)} \sin(4\pi(x_2 - 0,9)^2) \right\} \quad (4.48a)$$

$$\text{Función RF: } f(x_1, x_2) = 24,234 \left\{ [(x_1 - 0,5)^2 + (x_2 - 0,5)^2] \cdot [0,75 - (x_1 - 0,5)^2 - (x_2 - 0,5)^2] \right\} \quad (4.48b)$$

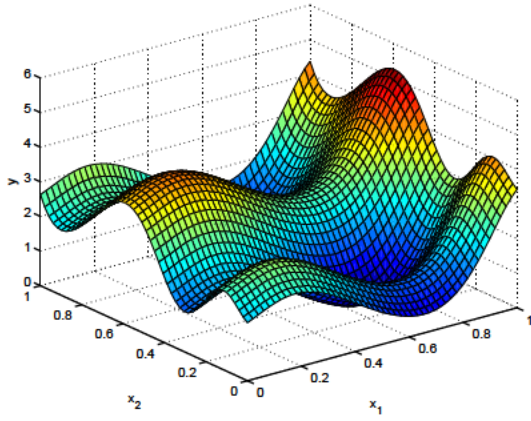
$$\text{Función SIF: } f(x_1, x_2) = 10,391 \{ (x_1 - 0,4)(x_2 - 0,6) + 0,36 \} \quad (4.48c)$$

$$\text{Función CIF: } f(x_1, x_2) = 1,9 \left\{ 1,35 + e^{(x_1-x_2)} \sin(13(x_1 - 0,6)^2) \sin(7x_2) \right\} \quad (4.48d)$$

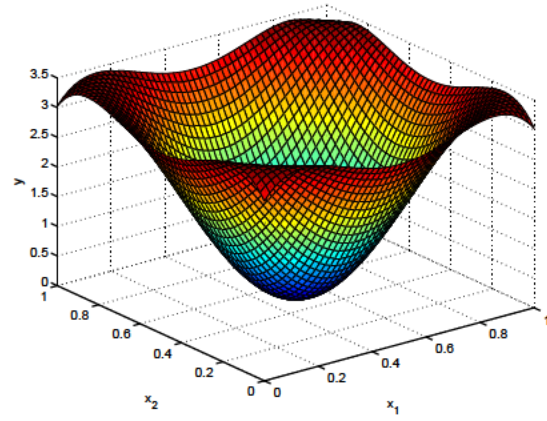
$$\text{Función SIF: } f(x_1, x_2, x_3) = \frac{1}{1 + \exp[-e^{x_1} + (x_2 - 0,5)^2 + 3 \sin(\pi x_3)] + 2x_1 \sin(2\pi x_2)} \quad (4.48e)$$

El conjunto de entrenamiento de las cuatro primeras funciones consta de 225 muestras distribuidas en una rejilla de puntos. Como conjunto de validación se utilizaron 10000 muestras generadas de manera uniforme en el dominio $[0, 1]^2$. Hay que tener en cuenta que los puntos de cada coordenada en el conjunto de entrenamiento en la Red neuronal de Chebychev deben corresponder a las raíces de polinomios de Chebychev de orden superior de la respectiva coordenada. En cambio, en el conjunto de validación no existe la necesidad de muestrear en las raíces de los polinomios de Chebychev y las muestras se encuentran equi-espaciadas. En la RBFNN existe libertad para generar el conjunto de entrenamiento. Por ello, en esta red neuronal se utilizó una rejilla uniforme de puntos equi-espaciados tanto en el conjunto de entrenamiento como en el de validación. Los puntos de validación se ubican en posiciones intermedias respecto a los puntos de entrenamiento.

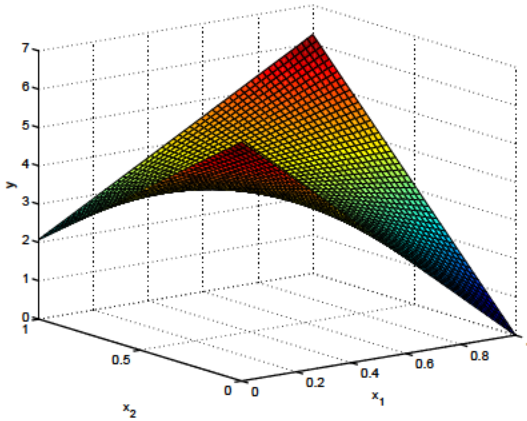
Para comprobar de forma más completa las habilidades de la nueva red neuronal se emplearon conjuntos de entrenamiento con ruido blanco gaussiano. Dicho ruido aditivo se calcula mediante la fórmula empleada en [Hwang et al., 1994], ver ecuación (4.49). Los valores de salida de las cuatro primeras funciones están ajustados para que su desviación estándar sea 1 en el dominio de trabajo. De este modo se consigue una relación señal a ruido (S/N) de aproximadamente valor 4 en escala lineal (6 dBs en escala logarítmica).



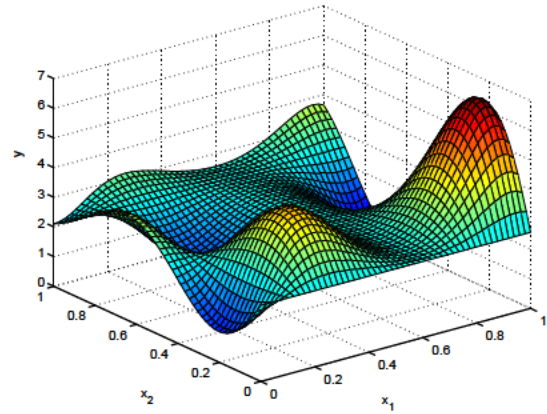
(a) Función AF (función aditiva).



(b) Función RF (función radial).



(c) Función SIF (función de interacción simple).



(d) Función CIF (función de interacción complicada).

Figura 4.9: Funciones utilizadas en el test numérico de aproximación de funciones multidimensionales.

$$f_{ent}(i) = f(x_1(i), x_2(i)) + 0,25\epsilon_1 \quad \epsilon_1 \sim N(0, 1) \quad i = 1, 2, \dots, 225 \quad (4.49)$$

La función con espacio de entrada tridimensional es muestreada a lo largo de 10 puntos en cada dimensión para dar lugar a un conjunto de entrenamiento de 1000 muestras. El conjunto de validación se compone por otros 1000 puntos generados por medio de un muestreo aleatorio uniforme. Así, se asegura que los datos de validación sean diferentes a los datos de entrenamiento. Al igual que en las cuatro primeras funciones, también se creó un conjunto de entrenamiento ruidoso con una señal a ruido de 10 dB. Las redes neuronales entrenadas con muestras ruidosas se testean con el conjunto de validación sin ruido en todos los casos. Todos los resultados mostrados con muestras ruidosas corresponden a la media de diez iteraciones.

Las Tablas 4.4, 4.5 y 4.6 muestran los niveles de NMSE de validación alcanzados por las redes neuronales para el conjunto de entrenamiento no ruidoso y diferentes números de neuronas. La Red neuronal de Chebychev entrenada con el algoritmo OLS alcanza un rendimiento superior a la RBFNN. La nueva red neuronal alcanza niveles de precisión muy elevados especialmente en las fun-

ciones AF, SIF y en la última función, superando ampliamente en estas tres funciones la eficiencia de la RBFNN. En el resto de funciones el rendimiento de la nueva red neuronal es ligeramente superior o similar al de la RBFNN. Este hecho se debe a que la función RF es similar a una gaussiana y por tanto la RBFNN es especialmente apta para aproximar esta función. La función CIF contiene valles y picos de modo que su aproximación es complicada para cualquier red neuronal. En general las funciones AF y SIF son más sencillas de aproximar que el resto de funciones debido a sus lentas variaciones. Los peores niveles de error se obtienen con la Red neuronal de Chebychev con selección consecutiva de neuronas. Claramente, la ausencia de un algoritmo de entrenamiento que elija las mejores neuronas provoca que la precisión no aumente conforme crece el tamaño neuronal. Los resultados de validación obtenidos con conjuntos de entrenamiento ruidosos se exponen en las Tablas 4.7, 4.8 y 4.9. De nuevo los mejores resultados corresponden a la nueva red neuronal entrenada con el algoritmo OLS. En esta ocasión la aproximación de las funciones se torna en más difícil. Incluso, con la nueva Red neuronal de Chebychev aparece un problema de sobre-entrenamiento leve en la última función. Para todas las redes neuronales el aumento del número de neuronas no conlleva un descenso claro del error de validación. Este hecho se debe a la diferencia entre el conjunto de entrenamiento ruidoso y el conjunto de validación libre de ruido. Este problema no es importante en la aproximación de las funciones de Green ya que los datos de salida no poseerán ruido en ningún caso. Las funciones de salida son conocidas con exactitud. Los datos presentados con funciones multidimensionales sugieren que la Red neuronal de Chebychev será capaz de aproximar correctamente las funciones de Green. Es razonable pensar que la nueva red neuronal conseguirá reducir el tamaño neuronal, para alcanzar la misma precisión que las RBFNNs, en el problema de aproximación de las funciones de Green.

Tabla 4.4: NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada con el algoritmo OLS. En cada caso con datos de entrenamiento sin ruido.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$1,5430 \cdot 10^{-01}$	$1,0220 \cdot 10^{-14}$	$1,2056 \cdot 10^{-14}$	$1,2572 \cdot 10^{-14}$	$1,2708 \cdot 10^{-14}$	$1,2743 \cdot 10^{-14}$
RF	$6,2004 \cdot 10^{-01}$	$4,1621 \cdot 10^{-01}$	$2,7709 \cdot 10^{-01}$	$1,9066 \cdot 10^{-01}$	$1,3549 \cdot 10^{-01}$	$9,2824 \cdot 10^{-02}$
SIF	$1,3403 \cdot 10^{-01}$	$8,9790 \cdot 10^{-15}$	$9,2366 \cdot 10^{-15}$	$9,5390 \cdot 10^{-15}$	$9,3207 \cdot 10^{-15}$	$9,3160 \cdot 10^{-15}$
CIF	$6,0958 \cdot 10^{-01}$	$4,6572 \cdot 10^{-01}$	$3,2117 \cdot 10^{-01}$	$2,1415 \cdot 10^{-01}$	$1,6019 \cdot 10^{-01}$	$1,0070 \cdot 10^{-01}$
$F(x_1, x_2, x_3)$	$3,2810 \cdot 10^{-01}$	$4,1664 \cdot 10^{-02}$	$1,4139 \cdot 10^{-02}$	$5,8603 \cdot 10^{-03}$	$2,6367 \cdot 10^{-03}$	$1,7074 \cdot 10^{-03}$

Tabla 4.5: NMSE del conjunto de validación para la red RBFNN entrenada en cada caso con datos de entrenamiento sin ruido.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$2,1933 \cdot 10^{-01}$	$6,5758 \cdot 10^{-02}$	$1,8291 \cdot 10^{-02}$	$7,7038 \cdot 10^{-03}$	$8,4363 \cdot 10^{-04}$	$3,0495 \cdot 10^{-04}$
RF	$7,0065 \cdot 10^{-01}$	$4,2282 \cdot 10^{-01}$	$2,2370 \cdot 10^{-01}$	$1,7552 \cdot 10^{-01}$	$1,1908 \cdot 10^{-01}$	$9,5315 \cdot 10^{-02}$
SIF	$1,7197 \cdot 10^{-01}$	$5,5865 \cdot 10^{-02}$	$1,7334 \cdot 10^{-02}$	$6,9190 \cdot 10^{-03}$	$7,9029 \cdot 10^{-04}$	$3,3650 \cdot 10^{-04}$
CIF	$6,5611 \cdot 10^{-01}$	$4,0191 \cdot 10^{-01}$	$2,1625 \cdot 10^{-01}$	$1,6118 \cdot 10^{-01}$	$1,1099 \cdot 10^{-01}$	$8,6847 \cdot 10^{-02}$
$F(x_1, x_2, x_3)$	$7,1297 \cdot 10^{-01}$	$2,5867 \cdot 10^{-01}$	$1,4681 \cdot 10^{-01}$	$7,3306 \cdot 10^{-02}$	$4,8089 \cdot 10^{-02}$	$4,2635 \cdot 10^{-02}$

Podemos comparar el rendimiento de la Red neuronal de Chebychev con el obtenido con las redes neuronales de los trabajos de los que se han extraído las funciones multidimensionales anteriores [Hwang et al., 1994, Ma y Khorasani, 2005]. Las redes neuronales desarrolladas en los anteriores trabajos se basan en el uso polinomios de Hermite como funciones de activación. El algo-

Tabla 4.6: NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada sin el algoritmo OLS. En cada caso se emplean datos de entrenamiento sin ruido.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$7,7152 \cdot 10^{-01}$	$7,7152 \cdot 10^{-01}$	$7,7152 \cdot 10^{-01}$	$7,7152 \cdot 10^{-01}$	$7,7152 \cdot 10^{-01}$	$7,7152 \cdot 10^{-01}$
RF	$8,4122 \cdot 10^{-01}$	$8,3833 \cdot 10^{-01}$	$8,3833 \cdot 10^{-01}$	$4,8728 \cdot 10^{-01}$	$4,8013 \cdot 10^{-01}$	$4,8013 \cdot 10^{-01}$
SIF	$8,1086 \cdot 10^{-01}$	$8,1086 \cdot 10^{-01}$	$8,1086 \cdot 10^{-01}$	$8,1086 \cdot 10^{-01}$	$8,1086 \cdot 10^{-01}$	$8,1086 \cdot 10^{-01}$
CIF	$7,7512 \cdot 10^{-01}$	$7,7234 \cdot 10^{-01}$	$7,7234 \cdot 10^{-01}$	$6,3196 \cdot 10^{-01}$	$6,3451 \cdot 10^{-01}$	$6,3451 \cdot 10^{-01}$
$F(x_1, x_2, x_3)$	$9,6386 \cdot 10^{-01}$	$9,6383 \cdot 10^{-01}$	$8,1772 \cdot 10^{-01}$	$8,1772 \cdot 10^{-01}$	$8,1748 \cdot 10^{-01}$	$8,1748 \cdot 10^{-01}$

Tabla 4.7: NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada con el algoritmo OLS. En cada caso con datos de entrenamiento con ruido blanco gaussiano.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$3,0802 \cdot 10^{-01}$	$2,5715 \cdot 10^{-01}$	$2,3643 \cdot 10^{-01}$	$2,2950 \cdot 10^{-01}$	$2,1591 \cdot 10^{-01}$	$2,0252 \cdot 10^{-01}$
RF	$6,4154 \cdot 10^{-01}$	$4,5957 \cdot 10^{-01}$	$3,4466 \cdot 10^{-01}$	$2,8648 \cdot 10^{-01}$	$2,4187 \cdot 10^{-01}$	$2,2721 \cdot 10^{-01}$
SIF	$1,3825 \cdot 10^{-01}$	$9,8421 \cdot 10^{-02}$	$1,2933 \cdot 10^{-01}$	$1,4447 \cdot 10^{-01}$	$1,5119 \cdot 10^{-01}$	$1,7248 \cdot 10^{-01}$
CIF	$6,1036 \cdot 10^{-01}$	$4,8064 \cdot 10^{-01}$	$3,2410 \cdot 10^{-01}$	$2,2030 \cdot 10^{-01}$	$1,8573 \cdot 10^{-01}$	$1,7664 \cdot 10^{-01}$
$F(x_1, x_2, x_3)$	$3,2849 \cdot 10^{-01}$	$6,5927 \cdot 10^{-02}$	$9,1870 \cdot 10^{-02}$	$1,1428 \cdot 10^{-01}$	$1,3428 \cdot 10^{-01}$	$1,4579 \cdot 10^{-01}$

ritmo desarrollado en [Ma y Khorasani, 2005] supera a los resultados de [Hwang et al., 1994] y alcanza niveles de error similares a los conseguidos con la nueva red neuronal. La comparación solo es posible con la situación de entrenamiento con muestras ruidosas ya que los resultados sin ruido gaussiano no se muestran en el mencionado trabajo. Las redes basadas en polinomios de Hermite en [Hwang et al., 1994] y [Ma y Khorasani, 2005] requieren de tamaños neuronales reducidos para aproximar la función objetivo. Sin embargo, los algoritmos empleados usan algoritmos de optimización del error basados en el uso del gradiente. Estos algoritmos pueden ser poco efectivos cuando el conjunto de entrenamiento consta de miles de muestras como en nuestro caso. En [Ma y Khorasani, 2005], el algoritmo de entrenamiento de la red neuronal de Hermite es mejorado frente a otros algoritmos, tanto en la precisión de la aproximación, como en el tiempo de computación. Sin embargo, no se muestran resultados temporales ni el número de operaciones del algoritmo. Además, los polinomios de Hermite se definen de forma recursiva. De este modo, incluso un número pequeño de neuronas puede producir que el cálculo de los valores de salida sea lento. En cambio, la nueva red neuronal carece de este problema gracias a la definición trigonométrica de los polinomios de Chebychev.

Tabla 4.8: NMSE del conjunto de validación para la red RBFNN entrenada en cada caso con datos de entrenamiento con ruido blanco gaussiano.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$3,3566 \cdot 10^{-01}$	$3,3489 \cdot 10^{-01}$	$3,8717 \cdot 10^{-01}$	$4,3249 \cdot 10^{-01}$	$4,6095 \cdot 10^{-01}$	$4,8409 \cdot 10^{-01}$
RF	$8,4874 \cdot 10^{-01}$	$6,3812 \cdot 10^{-01}$	$5,5307 \cdot 10^{-01}$	$5,0291 \cdot 10^{-01}$	$4,8187 \cdot 10^{-01}$	$4,9433 \cdot 10^{-01}$
SIF	$2,1716 \cdot 10^{-01}$	$1,0619 \cdot 10^{-01}$	$1,0595 \cdot 10^{-01}$	$1,3903 \cdot 10^{-01}$	$1,6488 \cdot 10^{-01}$	$2,0609 \cdot 10^{-01}$
CIF	$8,0096 \cdot 10^{-01}$	$5,4685 \cdot 10^{-01}$	$4,1822 \cdot 10^{-01}$	$3,1936 \cdot 10^{-01}$	$2,4356 \cdot 10^{-01}$	$2,4084 \cdot 10^{-01}$
$F(x_1, x_2, x_3)$	$7,0407 \cdot 10^{-01}$	$2,6896 \cdot 10^{-01}$	$2,2948 \cdot 10^{-01}$	$1,7794 \cdot 10^{-01}$	$1,7353 \cdot 10^{-01}$	$1,7119 \cdot 10^{-01}$

Tabla 4.9: NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada sin el algoritmo OLS. En cada caso se emplean datos de entrenamiento con ruido blanco gaussiano.

Función	5 neuronas	10 neuronas	15 neuronas	20 neuronas	25 neuronas	30 neuronas
AF	$7,8644 \cdot 10^{-01}$	$7,8440 \cdot 10^{-01}$	$7,8978 \cdot 10^{-01}$	$7,8185 \cdot 10^{-01}$	$7,8711 \cdot 10^{-01}$	$7,8763 \cdot 10^{-01}$
RF	$8,5253 \cdot 10^{-01}$	$8,4465 \cdot 10^{-01}$	$8,4185 \cdot 10^{-01}$	$5,2454 \cdot 10^{-01}$	$5,1315 \cdot 10^{-01}$	$5,0564 \cdot 10^{-01}$
SIF	$8,1162 \cdot 10^{-01}$	$8,1029 \cdot 10^{-01}$	$8,1213 \cdot 10^{-01}$	$8,0887 \cdot 10^{-01}$	$8,1523 \cdot 10^{-01}$	$8,1862 \cdot 10^{-01}$
CIF	$7,7622 \cdot 10^{-01}$	$7,7425 \cdot 10^{-01}$	$7,7479 \cdot 10^{-01}$	$6,3448 \cdot 10^{-01}$	$6,3946 \cdot 10^{-01}$	$6,4176 \cdot 10^{-01}$
$F(x_1, x_2, x_3)$	$9,6400 \cdot 10^{-01}$	$9,6423 \cdot 10^{-01}$	$8,1879 \cdot 10^{-01}$	$8,1885 \cdot 10^{-01}$	$8,1903 \cdot 10^{-01}$	$8,1869 \cdot 10^{-01}$

4.8. Capacidad de las Redes neuronales de Chebychev en la aproximación de las funciones de Green

En la Sección 4.5 se han determinado los fundamentos teóricos de la nueva Red neuronal de Chebychev. A continuación, en la Sección 4.7, se ha comprobado la capacidad de las Redes neuronales de Chebychev en la aproximación de diversas funciones multidimensionales. El siguiente paso consiste en aplicar las Redes neuronales de Chebychev entrenadas con el algoritmo OLS en la aproximación de las funciones de Green del medio multicapa. Al igual que en las RBFNNs se han aplicado una serie de tests numéricos para determinar la habilidad de las Redes neuronales de Chebychev en esta tarea de aproximación. Siguiendo el procedimiento de la Sección 3.9.2 los tests numéricos se llevaron a cabo en dos cavidades distintas y en diferentes regiones del espacio de entrada. Las zonas observación y fuente en la nueva Red neuronal de Chebychev son diferentes a las empleadas en las RBFNNs. Por lo tanto, en primer lugar explicaremos el modo y la forma en la que se generan dichas zonas.

4.8.1. Zonas observación y fuente en las Redes neuronales de Chebychev

Las Redes neuronales de Chebychev poseen diversas peculiaridades que determinan la forma de las zonas en las que se divide el espacio de entrada. El rango original de la función a aproximar debe ser trasladado al conjunto compacto $[-1, 1]^n$ en el caso de un espacio multidimensional de dimensión n . Además, para conseguir que los regresores de la matriz \mathbf{P} sean ortogonales es necesario muestrear en todas las raíces del polinomio de orden superior de cada coordenada. Esta característica fuerza a que la forma de las zonas observación sea cuadrada o rectangular. Este tipo de zonas permiten que la rejilla de puntos formada por las raíces de los polinomios de orden superior sea completada de forma total. Si tan solo se eliminase una de las raíces en una determinada coordenada para producir zonas distintas de las mencionadas, aparecerían casos de regresores no ortogonales.

Imaginemos que se desea llenar una zona circular de puntos observación (dos dimensiones). Para hacerlo de forma eficiente es necesario que los puntos de entrenamiento puedan situarse sobre las circunferencias que componen la zona, desde la exterior hasta la más interior. Esta forma de distribución de los puntos es muy difícil de conseguir cuando los valores en la abscisa y la ordenada no pueden ser fijados libremente (como ocurre en las Redes neuronales de Chebychev). Una posible solución consiste en sobre-muestrear cada coordenada y seleccionar únicamente aquellos puntos

situados en el interior de la zona circular. Sin embargo, este procedimiento produciría la pérdida de la ortogonalidad entre los regresores. Como se vio en la Sección 3.9.2, la mejora de la precisión debida a las zonas circulares es mucho menor que la producida por la técnica de eliminación de la singularidad basada en la extracción de dos capas de imágenes espaciales. La utilización de zonas cuadradas de forma única tendrá una incidencia relativamente pequeña en la exactitud de la aproximación. Las zonas cuadradas utilizadas en [García et al., 2006b], mostradas en la Figura 3.51, se dividen en dos tipos diferentes. La primera zona observación es puramente cuadrada. El resto de zonas, llamadas zonas de observación intermedias, tienen la forma geométrica de un cuadrado con un espacio vacío interior correspondiente a un cuadrado de menor área, es decir, tienen forma de corona rectangular o anillo cuadrado. La mismas razones aplicadas en las zonas circulares imponen el descarte de este segundo tipo de zonas cuadradas. La generación de un conjunto de entrenamiento para una corona rectangular implica la eliminación de los puntos situados en el espacio interior, que debe quedar vacío, haciendo imposible el uso de este tipo de figuras geométricas como zonas observación. Para dividir de forma correcta el espacio de entrada se debe recurrir a una división más profunda que la realizada con las RBFNNs. Cada una de las antiguas zonas cuadradas se debe dividir en cuatro sub-zonas diferentes. Cada una de estas sub-zonas se llenará con una rejilla regular de puntos. En adelante, denominaremos a dichas zonas: zona arriba, zona abajo, zona derecha y zona izquierda. Esta división es mucho más densa que la realizada en el Capítulo anterior, de modo que se compensará ampliamente el hecho de utilizar zonas cuadradas en lugar de zonas circulares.

En la Figura 4.10 se dibujan las nuevas zonas cuadradas, adecuadas para el entrenamiento de las Redes neuronales de Chebychev. Hemos denominado x_{or}/a e y_{or}/b a las coordenadas originales normalizadas de la cavidad. Podemos apreciar que los puntos exteriores no se sitúan en la línea de separación de las zonas. Este hecho se debe a que las raíces de los polinomios de Chebychev nunca toman los valores de los extremos -1 o 1 del rango de trabajo. Para validar la red neuronal entrenada se puede muestrear la zona con total libertad. De este modo, en cada conjunto de validación se generarán puntos en las partes exteriores de la zona para que la validación sea completa y exigente. En estos puntos, la Red neuronal de Chebychev extrapola a partir de los valores aprendidos durante el entrenamiento.

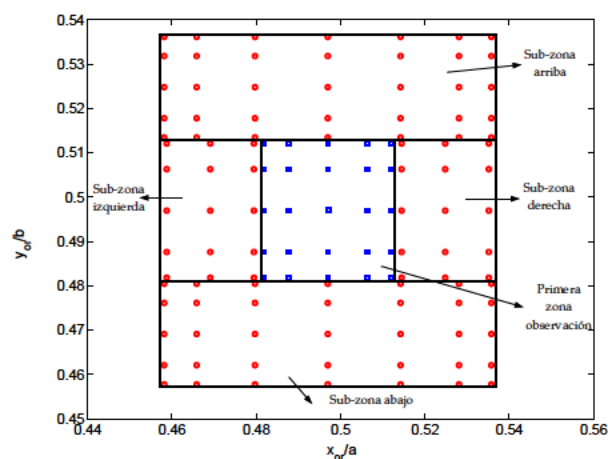


Figura 4.10: Nuevas sub-zonas observación cuadradas generadas para el entrenamiento de las Redes neuronales de Chebychev.

Una vez producido el conjunto de entrenamiento o validación es necesario trasladar el rango de cada una de las coordenadas originales $(x_{or}/a, y_{or}/b)$ al rango de trabajo $[-1, 1]^2$. En la Figura 4.11 se muestra la distribución de los puntos en la primera zona observación. Asimismo, en la Figura 4.12 se dibujan los puntos correspondientes a las zonas arriba y derecha. Obviamente, cuanto mayor es el número de puntos en una coordenada mayor será la cercanía de los puntos exteriores a los valores límite -1 y 1 . El número de puntos en las coordenadas de las zonas observación siempre es impar para garantizar que existe un punto que coincide con la abscisa u ordenada de la fuente. Así, las variaciones de las funciones de Green son expresadas de manera más completa.

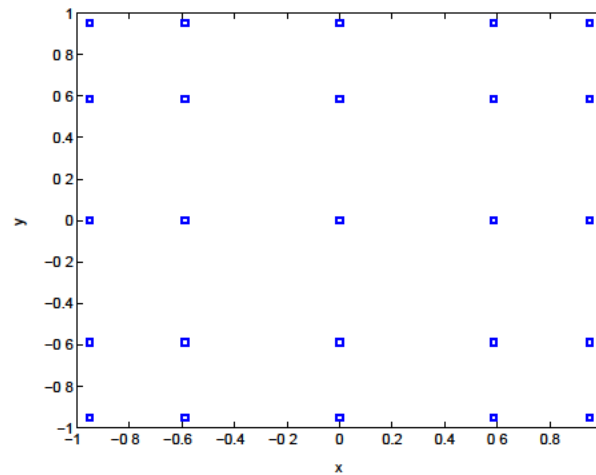


Figura 4.11: Primera zona observación en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev.

Las zonas fuente poseen dos formas básicas. La primera zona fuente es cuadrada mientras que las demás toman forma de "L". La primera zona fuente no presenta problemas en la distribución de los puntos de entrenamiento ya que el llenado se realiza en forma de rejilla, ver Figura 4.13. Al igual que en las zonas observación circulares y las zonas observación intermedias, la forma de "L" requeriría que se descartasen puntos de la rejilla bidimensional que forman todas las raíces. De este modo, para completar la rejilla es necesario conseguir zonas puramente rectangulares. Por ello, cada zona en forma de "L" se divide en dos sub-zonas como se aprecia en la Figura 4.14. A estas dos sub-zonas se les denominará zona fuente vertical y zona fuente horizontal. En todas las zonas fuente se muestra la localización de una serie de puntos de validación. Como se ha comentado para el caso de zonas observación, estos puntos se ubican en las líneas de separación para validar la aproximación de forma eficaz. En la Figura 4.14 la sub-zona vertical es más pequeña que la sub-zona horizontal, sin embargo, si se desea, se puede fijar una sub-zona vertical de mayor tamaño que la sub-zona horizontal. Ambas soluciones conducen a precisiones similares en la aproximación de las funciones de Green. En aquella sub-zona más grande existirá una menor densidad de puntos de entrenamiento y en general se deberá emplear un número ligeramente superior de neuronas para alcanzar la precisión.

El traslado de todos los puntos de entrenamiento y validación de las zonas fuente al conjunto de trabajo $[-1, 1]^2$ produce las distribuciones de puntos mostrados en las Figuras 4.15.

El algoritmo de división del espacio de entrada desarrollado en Sección 3.8 se aplica con ligeras

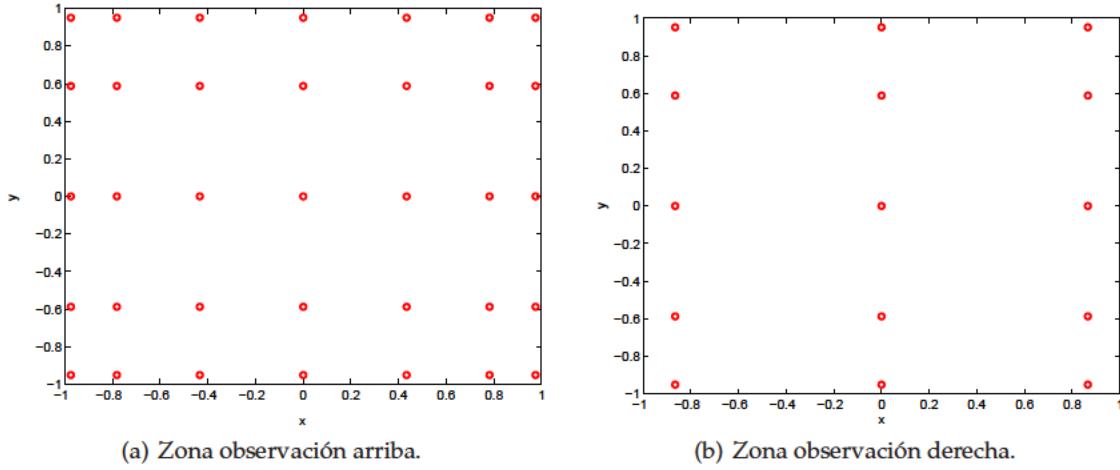


Figura 4.12: Zona observación arriba y derecha en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev.

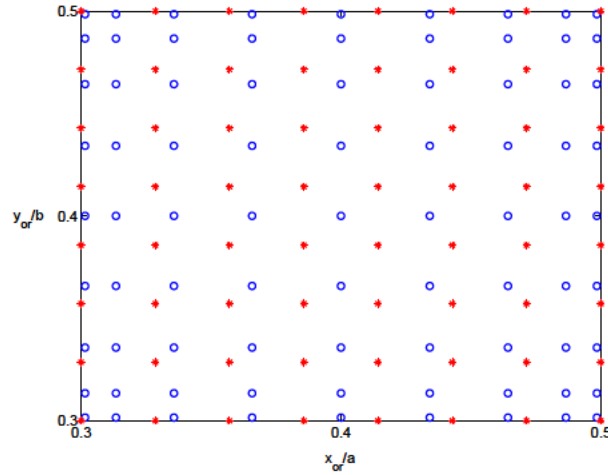


Figura 4.13: Primera zona fuente generada para el entrenamiento de las Redes neuronales de Chebychev. Los círculos son puntos de entrenamiento y los asteriscos son puntos de validación.

variaciones. Para obtener una determinada zona observación es necesario que la aproximación sea exitosa en las cuatro sub-zonas que la componen. Del mismo modo, para generar una zona fuente la aproximación debe superar los límites del error en las dos sub-zonas vertical y horizontal. Si en alguna de las sub-zonas la aproximación falla, el algoritmo procede al aumento del número de muestras. En caso de sobrepasar el tamaño máximo del conjunto de entrenamiento se reduce el área de la zona para dar lugar a un nuevo ciclo.

4.8.2. Aproximación de las funciones de Green

Las cavidades utilizadas en los presentes tests numéricos son las mismas que aquellas empleadas en los tests correspondientes a las RBFNNs mostrados en la Sección 3.9.2. También existen dos zonas fuente, una alejada de las paredes y otra situada cerca de las paredes. Al igual que en el test de las

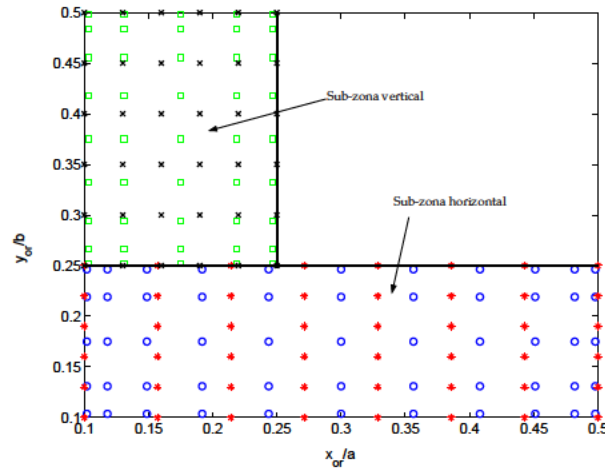


Figura 4.14: Nuevas sub-zonas fuente generadas para el entrenamiento de las Redes neuronales de Chebychev. Los círculos y los cuadrados son puntos de entrenamiento, los asteriscos y los cuadrados son puntos de validación.

RBFNNs se comprueba la capacidad de aproximación en dos zonas observación cercanas a la fuente. Una zona posee un área normalizada igual a $7,9 \cdot 10^{-4}$ mientras que la posee un área normalizada menor igual a $7,9 \cdot 10^{-6}$. Estas zonas son iguales a las zonas cuadradas utilizadas en el test con las RBFNNs. En esta ocasión, la zona fuente en forma de "L" cercana a las paredes debe ser dividida en dos sub-zonas, una de tipo horizontal y otra de tipo vertical. Debido a la ausencia de zonas observación de tipo circular, en cada zona fuente se comparan únicamente los resultados obtenidos con las dos técnicas de eliminación de la singularidad. De este modo, para todas las figuras posteriores, en la primera zona fuente los resultados obtenidos con la técnica de multiplicación por (ρ) se muestran con círculos y los correspondientes a la técnica de extracción de dos capas de imágenes con cruces. En la zona fuente situada cerca de las paredes, se comparan además la dos sub-zonas horizontal y vertical. Por un lado, los resultados para la técnica de multiplicación por (ρ) se muestran en la sub-zona vertical con diamantes y en la sub-zona horizontal con círculos. Por otro lado, los niveles de error correspondientes a la técnica de extracción de dos capas se dibujan con cruces en la sub-zona fuente horizontal y con asteriscos en la sub-zona vertical.

Se ha realizado un primer test con las mismas zonas fuente empleadas en el caso de las RBFNNs, ver Figura 3.9.2 en la Sección 3.9.2. Así, la primera zona fuente posee un límite inferior igual a $x/a = y/b = 0,40$. La zona fuente cercana a las paredes tiene un límite superior igual a $x/a = y/b = 0,015$. El límite inferior en esta zona era igual a $x/a = y/b = 0,001$ en el test realizado con las RBFNNs. En las presentes pruebas reducimos el límite hasta cero para comprobar la capacidad de aproximación de las Redes neuronales de Chebychev de forma más completa. Además, en cada prueba la razón entre el número de datos de validación y entrenamiento es superior al ratio empleado en los tests con las RBFNNs. Por ello, la comprobación de la capacidad de generalización es más intensa en las Redes neuronales de Chebychev. De esta forma, se refuerza la verificación de la capacidad de extrapolación en los límites de cada zona observación y fuente.

En las Figuras 4.16(a) y 4.16(b) se observa el error cuadrático medio correspondiente a la zona observación más grande en la cavidad 1, para la zona fuente alejada de las paredes y para las sub-zonas

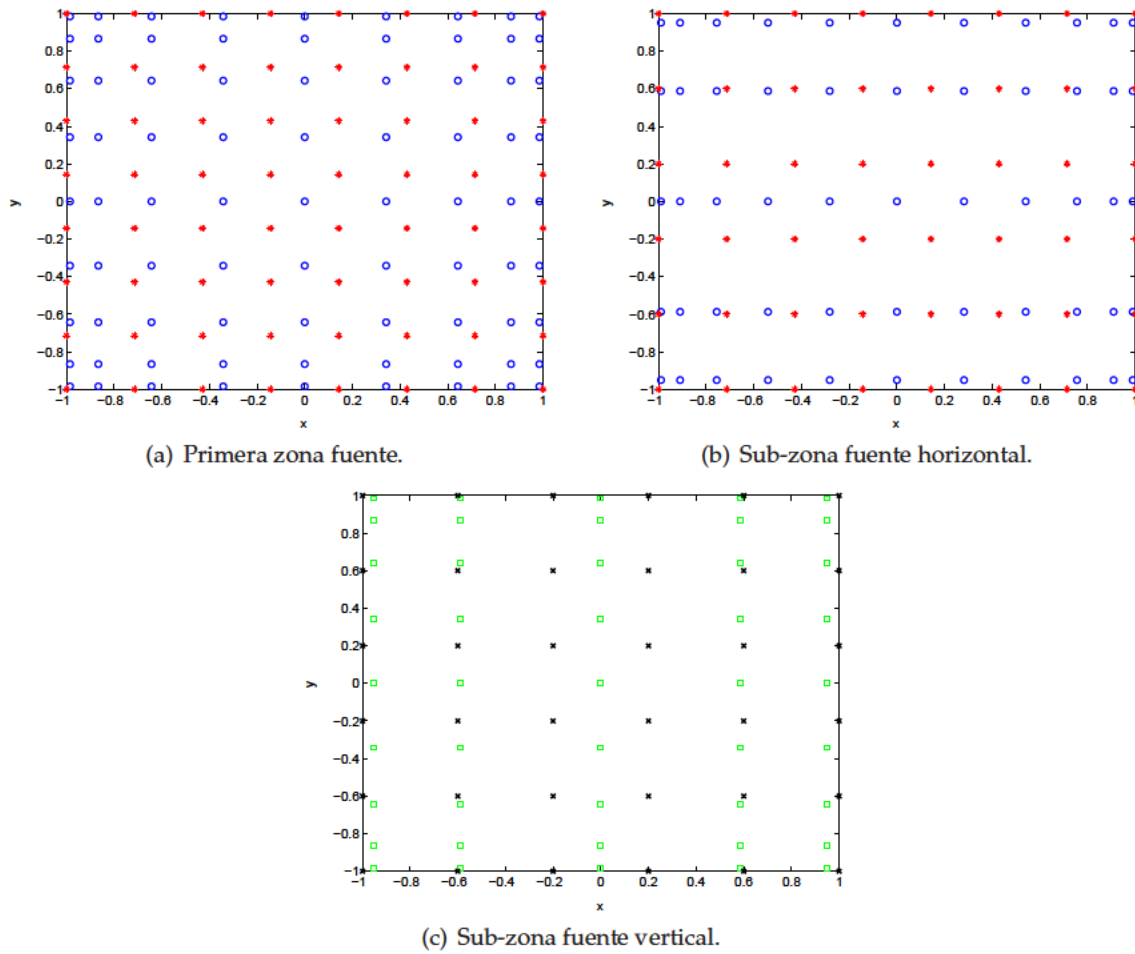
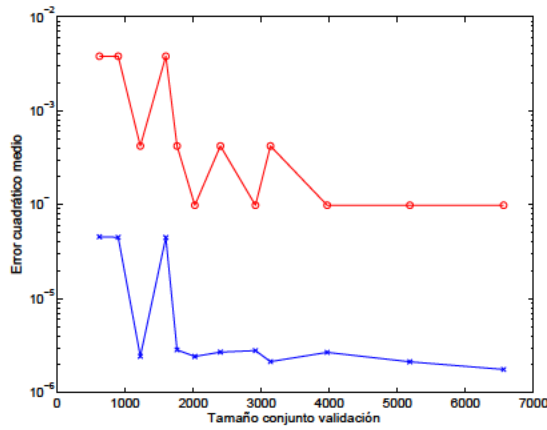


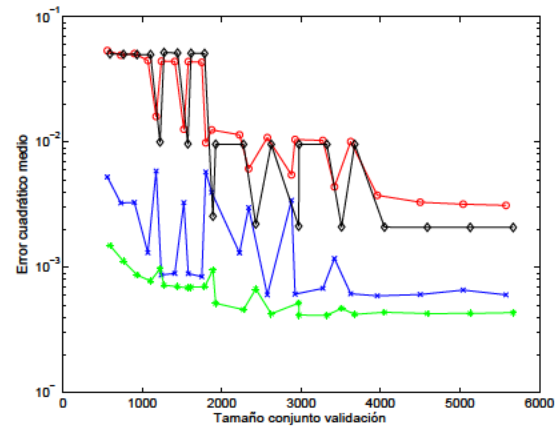
Figura 4.15: Primera zona fuente, sub-zona fuente vertical y sub-zona fuente derecha en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev. Los círculos son puntos de entrenamiento, los asteriscos son puntos de validación.

cercanas a las paredes respectivamente. El error en la aproximación converge a un valor mínimo para un valor de tan solo 30 neuronas para la zona alejada de las paredes. En la zona cercana a las paredes el número de neuronas debe ser ampliado a 250 para alcanzar un nivel de error mínimo. En las RBFNNs eran necesarias hasta 500 neuronas para alcanzar niveles de error semejantes a los conseguidos por las Redes neuronales de Chebychev (ver gráficas 3.54(a) y 3.56(a)). La reducción del tamaño neuronal es muy señalada.

Un resultado similar se observa en la zona observación de menor área. Para mayor brevedad se muestran únicamente los resultados correspondientes a la segunda cavidad en las Figuras 4.17(a) y 4.17(b). Se aprecia que en la zona alejada de las paredes la aproximación con la técnica de multiplicación por (ρ) alcanza una precisión parecida a la técnica de extracción de dos capas. Este hecho indica que las Redes neuronales de Chebychev son capaces de aproximar con habilidad similar tanto funciones de rápido crecimiento como de variación lenta. En los tests realizados con las RBFNNs el área tomada en las zonas fuente era reducida para que la aproximación con la técnica de multiplicación por (ρ) obtuviera cierta precisión. Con las Redes neuronales de Chebychev no es necesario recurrir a zonas fuente que posean áreas pequeñas. Así, en las siguientes pruebas el área de las zonas

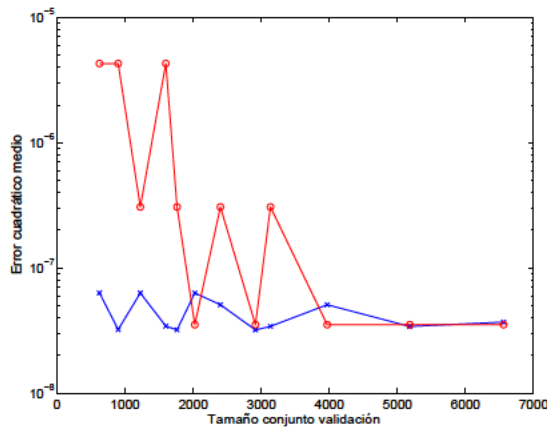


(a) Zona fuente alejada de las paredes con el límite inferior igual a 0,40. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.

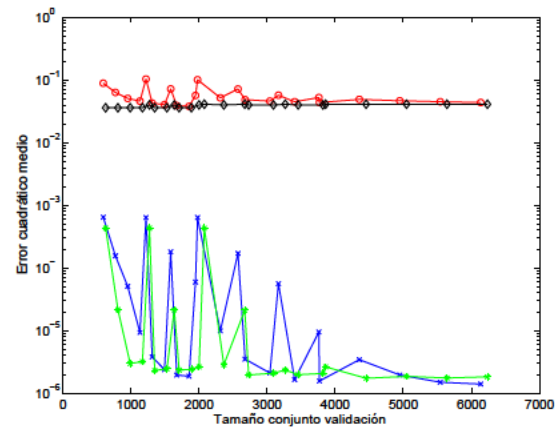


(b) Zona fuente situada cerca de las paredes con el límite inferior igual a 0,015. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.

Figura 4.16: Error cuadrático medio de los conjuntos de validación correspondientes a la zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. La cavidad corresponde a la estructura numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.



(a) Zona fuente alejada de las paredes con el límite inferior igual a 0,40. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.



(b) Zona fuente situada cerca de las paredes con el límite inferior igual a 0,015. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.

Figura 4.17: Error cuadrático medio de los conjuntos de validación correspondientes a la zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. La cavidad corresponde a la estructura numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

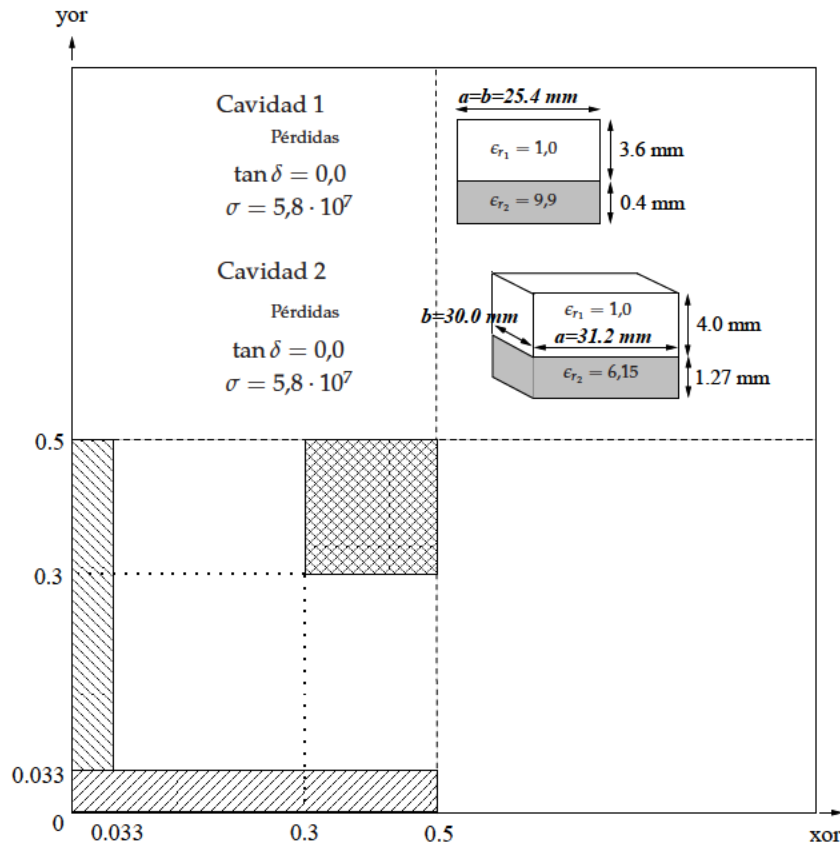
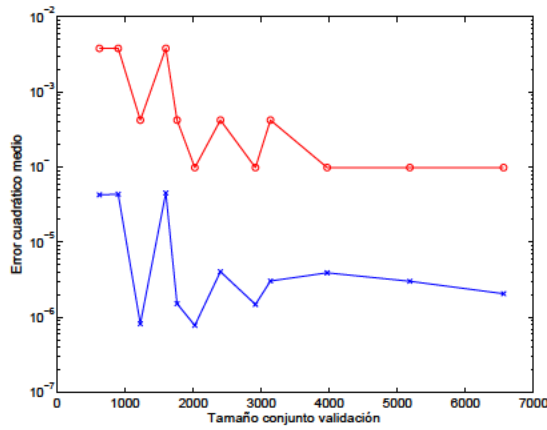


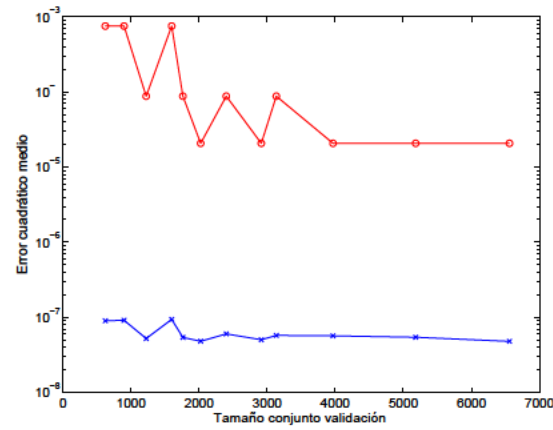
Figura 4.18: Zonas fuente y cavidades metálicas empleadas para testear la capacidad de aproximación de las funciones de Green por parte de las Redes neuronales de Chebychev. Las cavidades 1 y 2 son las mismas que fueron utilizadas en el Capítulo 3 como se observa en la Figura 3.53. La diferencia entre esta figura y la Figura 3.53 estriba en que ahora las zonas fuentes tienen un mayor área para mostrar la capacidad de aproximación de las Redes neuronales de Chebychev.

fuelle es ampliada tal como se observa en la Figura 4.18.

Con los nuevos límites se consiguen niveles de error de orden similar a los alcanzados en las zonas fuente de área reducida. En primer lugar se muestran los resultados correspondientes a la zona fuente alejada de las paredes para las dos zonas observación, Figuras 4.19 y 4.20. De nuevo, en ambos casos, únicamente son necesarias 30 neuronas para conseguir estabilidad en la precisión en cada uno de los conjuntos de entrenamiento y validación utilizados. En la zona observación más pequeña, la técnica de multiplicación por (ρ) obtiene una exactitud parecida a la conseguida por la extracción de dos capas de influencia. Este resultado confirma que si la zona observación tiene un tamaño relativamente grande las variaciones lentas son más fáciles de aproximar. En cambio, conforme se reduce la zona observación, la representación de las variaciones rápidas que se producen con la primera técnica de eliminación de la singularidad, mejora. De esta forma, las Redes neuronales de Chebychev pueden aproximar tanto un tipo de función de rápida variación como también de evolución lenta, siempre y cuando la densidad de puntos de entrenamiento sea suficiente. Por el contrario, las RBFNNs encuentran mayores dificultades para aproximar las funciones de crecimiento más rápido incluso si aumenta la densidad de puntos. Estos problemas en la aproximación se deben

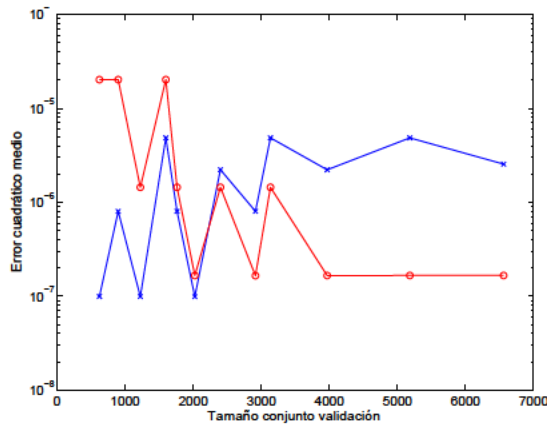


(a) Estructura multicapa numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.

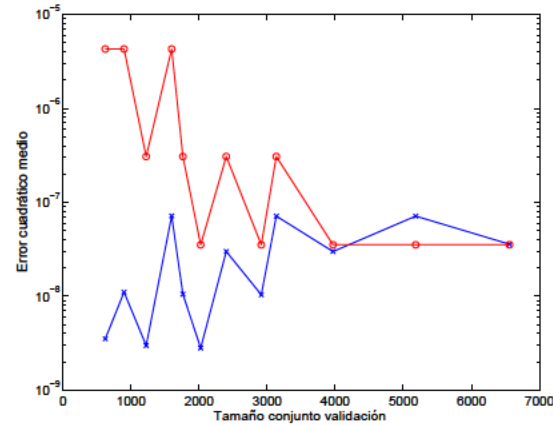


(b) Estructura multicapa numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

Figura 4.19: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.



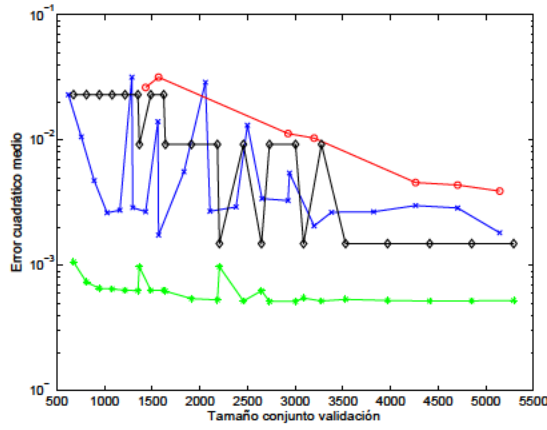
(a) Estructura multicapa numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.



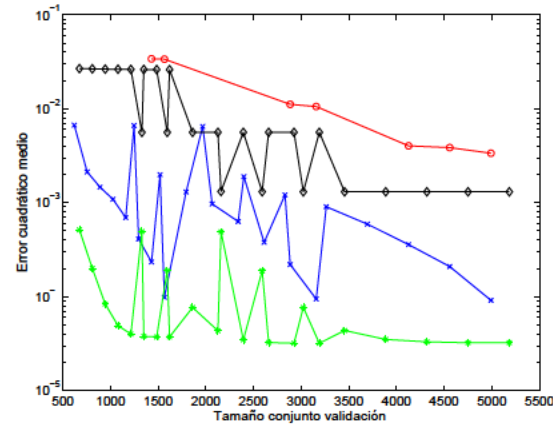
(b) Estructura multicapa numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

Figura 4.20: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.

a la obligada reducción de la varianza. Si este valor se reduce por debajo de un límite aparecerá con facilidad el problema del sobre-entrenamiento. Las Redes neuronales de Chebychev no presentan este inconveniente con asiduidad, ya que las funciones neuronales poseen valores significativos para cualquier valor de entrada. Por el contrario, las funciones gaussianas equivalen a deltas de Dirac situadas en los puntos de entrenamiento cuando la varianza toma valores muy reducidos.



(a) Estructura multicapa numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.

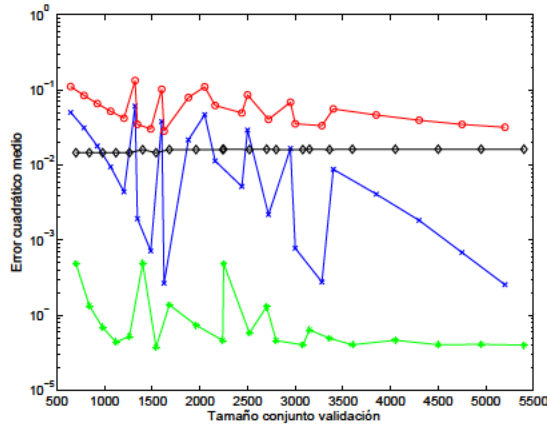


(b) Estructura multicapa numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

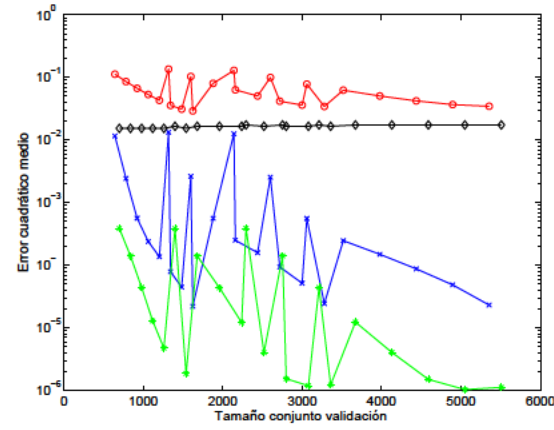
Figura 4.21: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.

Al igual que en la zona alejada de las paredes, la técnica de extracción de dos capas alcanza mayores niveles de precisión que la estrategia de multiplicación por (ρ) en las sub-zonas cercanas a los muros metálicos. Tanto en la zona observación de mayor tamaño (Figura 4.21) como en la zona observación pequeña (Figura 4.22) el número de neuronas máximo se cifra en 250. Este tamaño neuronal garantiza la convergencia del error de aproximación para los diferentes conjuntos de entrenamiento y validación en las sub-zonas cercanas a las paredes. La precisión es mayor en ambas cavidades en la sub-zona vertical. Este hecho se debe a que ésta presenta un área menor que la sub-zona horizontal. No obstante, la diferencia de precisión no impide que la sub-zona horizontal alcance asimismo bajos niveles de error en ambas cavidades y en las dos zonas observación. El método de análisis neuronal no se verá afectado por el diferente tamaño de una sub-zona respecto a otra, ya que se alcanza una precisión similar para el tamaño máximo de entrenamiento utilizado.

La estrategia de extracción de dos capas es más eficiente ya que permite la representación precisa de las funciones de Green en zonas observación de mayor tamaño. Así, se requerirán un número menor de zonas observación en la región MS situada en torno a la fuente. Además, cada una de estas zonas observación necesitará un número reducido de zonas fuente. El tiempo de selección y entrenamiento será menor que en el caso de la técnica de multiplicación por (ρ), aumentando de este modo el interés del método neuronal de análisis. El beneficio de la técnica de extracción de dos capas es doble ya que permite la recuperación exacta de la singularidad de las funciones de Green. En efecto, como se observa en las Figuras 4.23 y 4.26 los niveles de error una vez recuperada la singularidad son los mismos que los mostrados en las Figuras 4.19 y 4.22. En cambio, los niveles de error en la técnica de multiplicación por (ρ) son mucho mayores debido a la amplificación del error correspondiente a la aproximación cometida por la red neuronal al introducir la singularidad

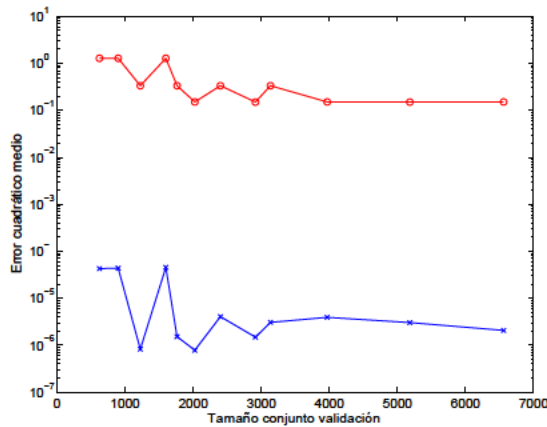


(a) Estructura multicapa numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.

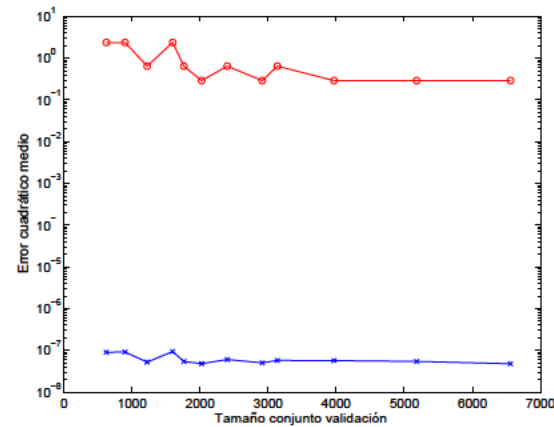


(b) Estructura multicapa numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

Figura 4.22: Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.



(a) Estructura multicapa numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.



(b) Estructura multicapa numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.

Figura 4.23: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.

(división por (ρ)). Al igual que en las RBFNNs el mecanismo de extracción de dos capas permite una eficiencia superior a la multiplicación por (ρ). Las Redes neuronales de Chebychev consiguen una precisión elevada con un número reducido de neuronas. Además, el peligro de sobre-entrenamiento aparece de forma menos común que en el caso de las RBFNNs.

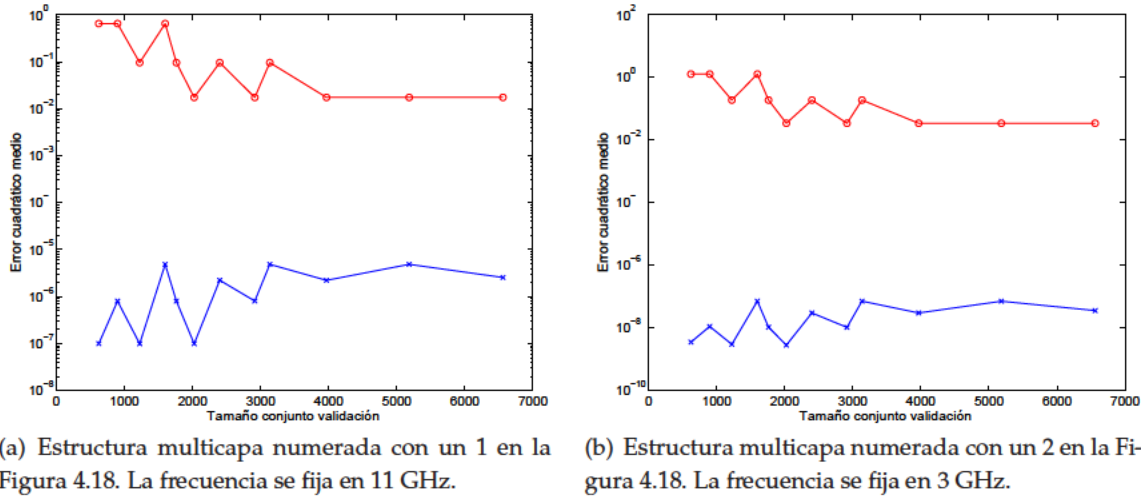


Figura 4.24: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.

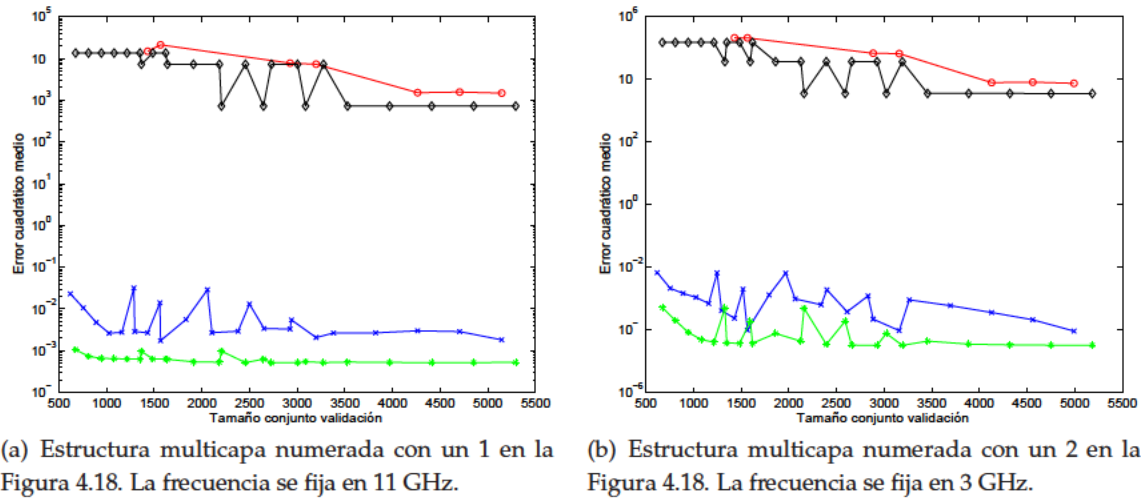


Figura 4.25: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.

En las figuras anteriores se han mostrado los niveles de error para una serie de distintos conjuntos de entrenamiento. El tamaño de dichos conjuntos varía desde pocos cientos de muestras hasta varios miles de datos. Como se explica en las Secciones 4.4 y 4.6 el nuevo método de entrenamiento OLS en las Redes neuronales de Chebychev permite una reducción drástica del número de operaciones

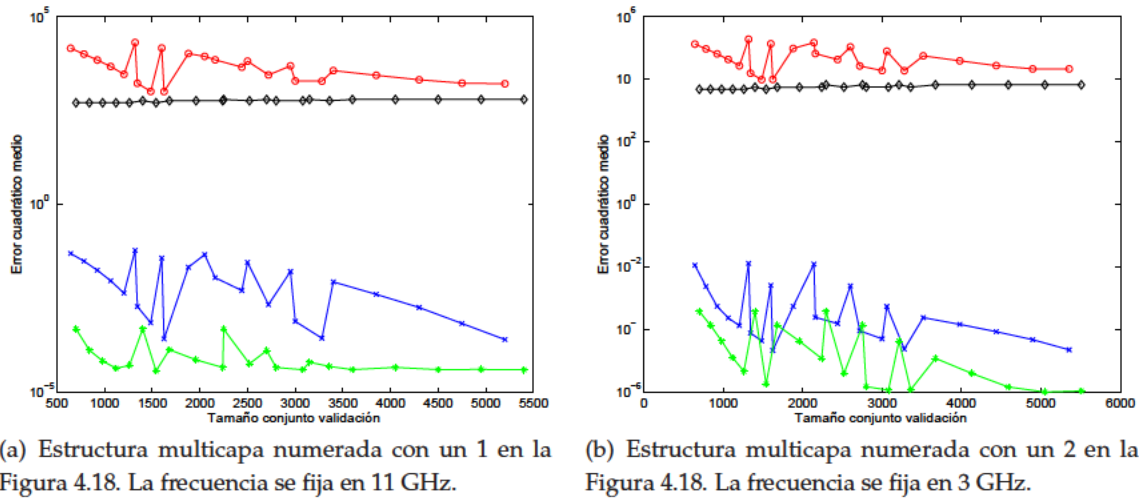


Figura 4.26: Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.

requeridas para diseñar la red neuronal. A continuación se expone el tiempo de entrenamiento para diferentes conjuntos de entrenamientos similares a los utilizados en las pruebas anteriores. En cada prueba se compara la versión rápida del algoritmo OLS explicado en la Sección 3.9.1, el algoritmo OLS para Redes neuronales de Chebychev con la fórmula de reducción del error sin simplificación (4.34) y el mismo algoritmo con la fórmula simplificada (4.36). En todos los casos se utiliza la primera zona fuente y una zona observación cuadrada cercana a la fuente. Estos valores no influyen en los resultados ya que se comparan únicamente resultados temporales. El número de neuronas se fijó en 500. El error de entrenamiento y validación se evalúa cada 100 neuronas.

En la Tabla 4.10 se aprecia la gran rapidez del algoritmo OLS en las Redes neuronales de Chebychev. El tiempo de entrenamiento es reducido de forma muy marcada frente a la versión más rápida del algoritmo OLS en las RBFNNs. El entrenamiento y validación en las redes es entre 15 y 25 veces más rápido. Además, hay que tener en cuenta que el conjunto de validación en las Redes neuronales de Chebychev es mucho mayor que en las RBFNNs. Esta característica es debida al deseo de validar de forma exigente las Redes neuronales de Chebychev y a la necesidad de colocar puntos de validación en la frontera entre zonas. Este hecho supone una ligera carga adicional en las pruebas realizadas con las Redes neuronales de Chebychev ya que el error se evaluó también en el conjunto de validación. El tamaño máximo de entrenamiento está fijado por la limitación en el tamaño de la matriz de regresores del algoritmo OLS en la RBFNN. Para acelerar este algoritmo es necesario almacenar dicha matriz. El algoritmo OLS en las Redes neuronales de Chebychev no posee esta limitación pudiendo seleccionar regresores en conjuntos de entrenamiento mayores con rapidez y fiabilidad.

Las dos versiones del algoritmo OLS en las Redes neuronales de Chebychev obtienen resultados

temporales muy similares. Esto es debido a que la versión simplificada requiere del cálculo de los pesos una vez seleccionados los regresores. Además, una parte considerable del tiempo de selección de las neuronas se emplea en el cálculo de los regresores.

Tabla 4.10: Tiempos de entrenamiento en segundos. El primer término es el número de puntos de entrenamiento para Redes neuronales de Chebychev y RBFNNs. El segundo término es el tamaño del conjunto de validación para las Redes neuronales de Chebychev. Entre paréntesis se muestra el tamaño del conjunto de validación para la RBFNN.

Número de fuentes	Número de puntos observación	Número de puntos frecuencia	Tamaño del conjunto de entrenamiento y validación	FOLS RBFNNs	OLS Chebychev sin simplif.	OLS Chebychev con simplif.
7	5	1	1225-1764 (1764)	18.8	1.30	1.28
7	7	1	2401-2916 (2916)	62.3	3.33	3.39
7	5	3	3675-7056 (3528)	138.8	8.93	9.16
7	5	5	6125-10584 (7056)	382.0	22.06	22.31
7	7	3	7203-11664 (5832)	511.9	29.15	29.87
7	7	5	12005-17496 (11664)	1428.2	77.00	76.72

4.9. Ventajas y desventajas de las Redes neuronales de Chebychev frente a las Redes neuronales de funciones de base radial

Las Redes neuronales de Chebychev entrenadas con el algoritmo OLS presentan una serie de ventajas y desventajas frente a las RBFNNs. Debido a que una ventaja en cierto aspecto puede acarrear alguna limitación o desventaja en otro proceso de la tarea de aproximación, se procede al comentario de las ventajas y desventajas de forma combinada:

- Como ya se ha mencionado en secciones anteriores, la utilización de las raíces del polinomio de Chebychev de orden superior en cada coordenada para la generación del conjunto de entrenamiento permite un rendimiento óptimo en el algoritmo OLS. Así, se obtienen redes mucho más compactas que las conseguidas con las RBFNNs. Este modo de operación conlleva la limitación del método propuesto a aplicaciones en las que la función objetivo puede ser muestreada en dichos puntos. Existen numerosas aplicaciones de las redes neuronales en las que los datos de entrenamiento no se pueden fijar libremente o no toman forma de rejilla uniforme. Este hecho es muy común en tareas de clasificación. Sin embargo, en el problema de la aproximación de las funciones de Green existe total libertad para generar los conjuntos de entrenamiento a conveniencia del algoritmo de diseño. De esta forma, la limitación del nuevo método neuronal no supone ningún obstáculo para su utilización en el análisis de circuitos de microondas utilizando técnica de Ecuación integral.
- El algoritmo de entrenamiento en las Redes neuronales de Chebychev es mucho más veloz que la versión más rápida del algoritmo OLS en las RBFNNs. Además, no es necesario guardar las actualizaciones de los regresores en cada paso. En el algoritmo OLS aplicado a las Redes neuronales de Chebychev hay que evaluar las contribuciones de los regresores a la energía de salida una sola vez al principio del procedimiento. Así, no existe ninguna limitación, debida a

la cantidad de memoria RAM del ordenador, en el tamaño del conjunto de entrenamiento. El entrenamiento de funciones muestreadas en decenas de miles de puntos se vuelve factible. Por tanto, el entrenamiento de las nuevas redes requiere muy pocos recursos computacionales en comparación con el algoritmo OLS en las RBFNNs.

- La limitación en el muestreo de la función objetivo obliga a la utilización de zonas observación cuadradas o rectangulares. Al contrario que en las RBFNNs, las nuevas Redes neuronales de Chebychev no pueden emplear zonas observación circulares, zonas observación con forma de corona rectangular o zonas en fuente en forma de "L". La división del espacio de entrada se torna más complicada. No obstante, la mayor velocidad en el entrenamiento permite que el tiempo total de diseño de las redes necesarias no sea superior al invertido en las RBFNNs.
- Las Redes neuronales de Chebychev neuronales no poseen ningún parámetro que deba ser ajustado de forma heurística. En las RBFNNs que utilizan funciones gaussianas, la varianza es un parámetro adicional que en ocasiones es fijado por el usuario. En nuestro problema, este inconveniente se solventó mediante la aplicación de un mecanismo que busca de forma iterativa un valor de varianza adecuado para la aproximación de las funciones de Green. Este procedimiento no es necesario en las Redes neuronales de Chebychev. Esta característica permite cierta simplificación en la división del espacio de entrada. Con las nuevas redes neuronales únicamente se debe proceder a realizar pasos alternativos de reducción del área de las zonas, y de llenado de dichas zonas hasta alcanzar la precisión fijada.

Podemos concluir que las principales desventajas de las Redes neuronales de Chebychev, entrenadas mediante el algoritmo OLS, frente a las RBFNNs son evitadas en el problema de la aproximación de las funciones de Green. En cambio, la clara ventaja consistente en la reducción del tamaño neuronal, va a servir para la aceleración del análisis de los circuitos de microondas impresos.

4.10. Análisis de circuitos encapsulados con las Redes neuronales de Chebychev

En la sección anterior se ha mostrado que las Redes neuronales de Chebychev consiguen una precisión similar a las RBFNNs en la aproximación de las funciones de Green. La gran ventaja de las nuevas redes estriba en el reducido tamaño neuronal que requieren para alcanzar dicha precisión. La sustitución de las funciones de Green por los valores de salida de las Redes neuronales de Chebychev producirá una gran reducción del tiempo de análisis de los circuitos encapsulados. Para comprobar esta destreza se ha procedido a aproximar las funciones de Green en las dos cavidades de trabajo numeradas con un 1 y un 2 en la Figura 4.18. Debido a la elevada exactitud alcanzada en la recuperación de la singularidad por parte de la técnica de extracción de dos capas se ha preferido esta técnica en la zona MS frente a la estrategia de multiplicación por (ρ) . En la zona DS situada lejos de la fuente se emplea la técnica de multiplicación por la distancia (ρ) . Una vez dividido el espacio de entrada y entrenadas todas las redes neuronales se pueden analizar todo tipo de filtros impresos. Por motivos comparativos se utilizarán los filtros paso banda analizados previamente con RBFNNs en la Sección 3.9.3. Al igual que en la RBFNNs el error máximo se fijó en un valor NMSE de 0,025.

A continuación se muestra la división del espacio de entrada llevada a cabo en la primera cavidad. En la Tabla 4.11 se exponen las zonas observación en las que queda dividida la región cercana a la fuente o región MS. Durante la selección de estas zonas se fijan los límites de la primera zona fuente. Gracias a la técnica de extracción de dos capas de imágenes espaciales y a la habilidad de las Redes neuronales de Chebychev el límite inferior de esta zona fuente es 0,3. En las RBFNNs esta zona poseía también el mismo tamaño, sin embargo, como se comprobará más adelante las Redes neuronales de Chebychev requieren menos neuronas. Por otra parte, la región DS, al igual que en el caso de las RBFNNs (ver Tabla 3.16), es dividida en cinco zonas.

Tabla 4.11: Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,20/\pi$ en la primera cavidad. Técnica de eliminación de la singularidad mediante la extracción de dos capas.

Zona	Límite inferior	Límite superior
1ª zona	Posición del punto fuente	$0,0500/\pi$
2ª zona	$0,0500/\pi$	$0,1250/\pi$
3ª zona	$0,1250/\pi$	$0,1625/\pi$
4ª zona	$0,1625/\pi$	$0,2000/\pi$

En las Tablas 4.13 y 4.14 se exponen el número de zonas fuente para cada una de las zonas observación de la región MS y DS respectivamente. Como se aprecia, cada sub-zona observación posee su propia división en zonas fuente. En todas ellas, así como en la zona observación más cercana a la fuente, el número de zonas fuente es reducido. Cada zona fuente distinta de la más alejada de las paredes se divide en dos sub-zonas vertical y horizontal. Por motivos de brevedad solo se muestra el número de zonas fuente, sus límites se pueden consultar en el Apéndice D.

Los tiempos empleados en la división de las zonas observación, fuente y frecuencia y el tiempo de entrenamiento se exhiben en la Tabla 4.15. En la coordenada frecuencial existe una única zona

Tabla 4.12: Zonas observación generadas en la región DS en la primera cavidad. Técnica de eliminación de la singularidad mediante la multiplicación por (ρ).

Zona	Límite inferior	Límite superior
1ª zona	$0,200/\pi$	$0,300/\pi$
2ª zona	$0,300/\pi$	$0,450/\pi$
3ª zona	$0,450/\pi$	$0,600/\pi$
4ª zona	$0,600/\pi$	$0,800/\pi$
5ª zona	$0,800/\pi$	1,000

Tabla 4.13: Número de zonas fuente correspondiente a las zonas observación de la región MS en la primera cavidad.

Tipo de zona	1ª zona observ.	2ª zona observ.	3ª zona observ.	4ª zona observ.
Primera zona	4	—	—	—
Arriba	—	5	6	5
Abajo	—	6	5	5
Derecha	—	4	4	4
Izquierda	—	4	6	5

situada entre 9 y 11 GHz. A pesar de la necesidad de dividir en zonas exclusivamente cuadradas o rectangulares el espacio de entrada, el tiempo requerido para seleccionar las diferentes sub-zonas es menor que en las RBFNNs. Este hecho se debe a dos causas. Por un lado, el entrenamiento de las Redes neuronales de Chebychev es muy veloz. Por otro lado, se ha aplicado un nuevo método de selección de zonas fuente en el que las combinaciones de puntos fuente no se exploran de forma consecutiva. Ahora, para un determinado número de puntos fuente en el lado menor del rectángulo se prueba en primer lugar una combinación de puntos fuente en los dos lados de gran densidad. Si hay éxito en esta primera prueba, se pasa a comprobar combinaciones de menor densidad. Si no hay éxito, empieza un nuevo ciclo en el que se incrementa el número de puntos fuente del lado menor. Este proceso descarta muchas de las combinaciones probadas en el algoritmo de exploración consecutiva utilizado con las RBFNNs.

En la segunda caja el espacio de entrada es asimismo dividido en un reducido número de zonas observación y fuente tal como se observa en las Tablas 4.16 y 4.19. La región MS requiere de cuatro zonas mientras que la región más alejada de la fuente necesita cinco zonas. Del mismo modo que en

Tabla 4.14: Número de zonas fuente correspondiente a las zonas observación de la región DS en la primera cavidad.

Tipo de zona	1ª zona observ.	2ª zona observ.	3ª zona observ.	4ª zona observ.	5ª zona observ.
Arriba	6	5	5	4	4
Abajo	5	5	4	3	3
Derecha	4	4	4	4	4
Izquierda	4	4	4	3	3

Tabla 4.15: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Técnica de extracción de dos capas en la región MS. Multiplicación por (ρ) en la región DS.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento Red neuronal de Chebychev	Tiempo total
Tiempo	1h 24m	24h 34m	1h 10m	21h 31m	46h 39m

la primera cavidad, el número de zonas y sub-zonas fuente en cada una de las zonas observación es pequeño. Las Redes neuronales de Chebychev permiten una adecuada división del espacio de entrada para alcanzar una gran precisión. Además, la estrategia de extracción de dos capas y de multiplicación por (ρ) en la región DS es también aprovechada por las nuevas redes tal como ocurría en las RBFNNs. Los límites de las distintas zonas y sub zonas fuente pueden consultarse en el Apéndice D.

Tabla 4.16: Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,20/\pi$ en la segunda cavidad. Técnica de eliminación de la singularidad mediante la extracción de dos capas.

Zona	Límite inferior	Límite superior
1ª zona	Posición del punto fuente	$0,0500/\pi$
2ª zona	$0,0500/\pi$	$0,1250/\pi$
3ª zona	$0,1250/\pi$	$0,1625/\pi$
4ª zona	$0,1625/\pi$	$0,2000/\pi$

Tabla 4.17: Zonas observación generadas en la región DS en la segunda cavidad. Técnica de eliminación de la singularidad mediante la multiplicación por (ρ).

Zona	Límite inferior	Límite superior
1ª zona	$0,200/\pi$	$0,450/\pi$
2ª zona	$0,450/\pi$	$0,600/\pi$
3ª zona	$0,600/\pi$	$0,750/\pi$
4ª zona	$0,750/\pi$	$0,850/\pi$
5ª zona	$0,850/\pi$	1,000

El tiempo de selección de las zonas anteriores se expone en la Tabla 4.20. Al igual que en la primera cavidad existe una única zona frecuencial situada ahora entre 1 y 3 GHz. En este caso, las Redes neuronales de Chebychev requieren un periodo de diseño similar al obtenido con las RBFNNs para un nivel de precisión similar. Esto es debido a que la aproximación de las funciones de Green es más sencilla en la segunda cavidad de forma que ambas técnicas ofrecen resultados similares en la precisión y tiempo invertido. Sin embargo, como se comprobará más adelante la Red neuronal de Chebychev alcanza la precisión requerida con muchas menos neuronas que la RBFNN.

Gracias a la rapidez del entrenamiento de las Redes neuronales de Chebychev y del nuevo proce-

Tabla 4.18: Número de zonas fuente correspondiente a las zonas observación de la región MS en la segunda cavidad.

Tipo de zona	1ª zona observ.	2ª zona observ.	3ª zona observ.	4ª zona observ.
Primera zona	4	—	—	—
Arriba	—	4	4	4
Abajo	—	5	4	4
Derecha	—	4	4	4
Izquierda	—	4	4	4

Tabla 4.19: Número de zonas fuente correspondiente a las zonas observación de la región DS en la segunda cavidad.

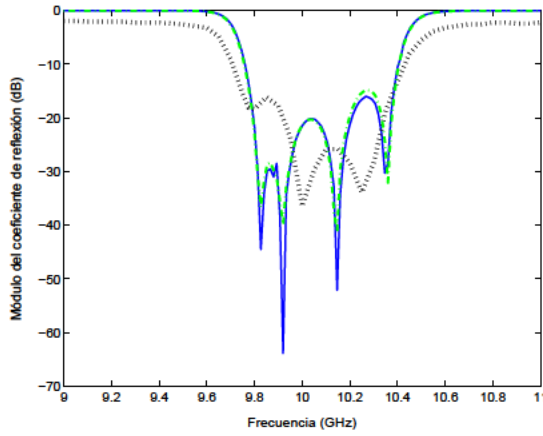
Tipo de zona	1ª zona observ.	2ª zona observ.	3ª zona observ.	4ª zona observ.	5ª zona observ.
Arriba	7	5	5	6	4
Abajo	5	4	4	3	2
Derecha	4	5	4	4	4
Izquierda	4	4	3	3	2

dimiento de generación de zonas fuente, el periodo de selección y entrenamiento, tanto en la primera como en la segunda cavidad, permite la utilización práctica del método de análisis neuronal. Pese a la división más profunda del espacio de entrada y el consiguiente aumento del número de redes neuronales, el tiempo total de diseño, respecto a las RBFNNs, es del mismo orden de magnitud en la segunda cavidad y mucho menor en la primera. Si se hubiera utilizado un método de entrenamiento en las Redes neuronales de Chebychev de rapidez similar a la conseguida por el algoritmo OLS en las RBFNNs, el periodo de diseño total en las Redes neuronales de Chebychev sería de cientos de horas en ambas cavidades convirtiendo en muy poco práctico al método neuronal.

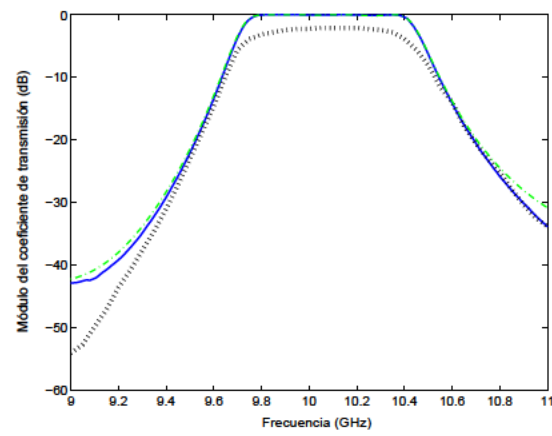
Tabla 4.20: Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Técnica de extracción de dos capas en la región MS. Multiplicación por (ρ) en la región DS.

	Zona observación	Zona fuente	Zona frecuencia	Entrenamiento Red neuronal de Chebychev	Tiempo total
Tiempo	40m	10h 5m	50m	37h 9m	48h 54m

Para comprobar la capacidad en el análisis de circuitos encapsulados del método neuronal basado en Redes neuronales de Chebychev se han analizado los mismos filtros previamente utilizados en la Sección 3.9.3. Las Figuras 4.27, 4.28 y 4.29 muestran los resultados para los filtros de líneas acopladas en la primera cavidad dibujados en las Figuras 3.37, 3.38 y 3.66. Se aprecia que en los tres casos el método neuronal consigue una gran precisión en todo el ancho de banda de trabajo. Incluso para el filtro de cinco polos que involucra acoplos muy críticos, el método neuronal sigue con exactitud las variaciones tanto en la banda de paso como en la banda eliminada (Figura 4.29). En este filtro existe un pequeño desplazamiento en frecuencia en la ubicación del cero de transmisión. No

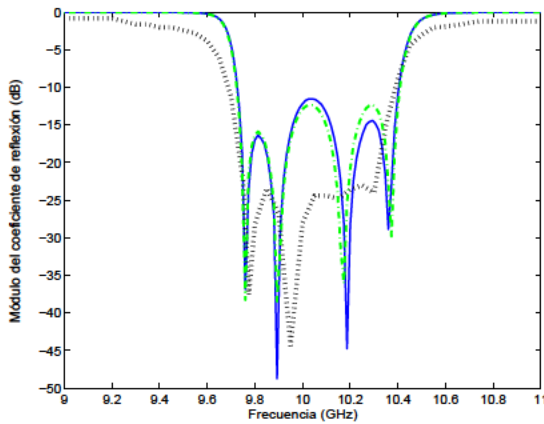


(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

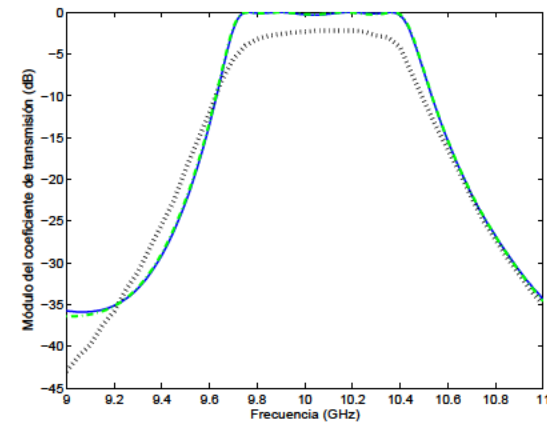


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.37.

Figura 4.27: Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

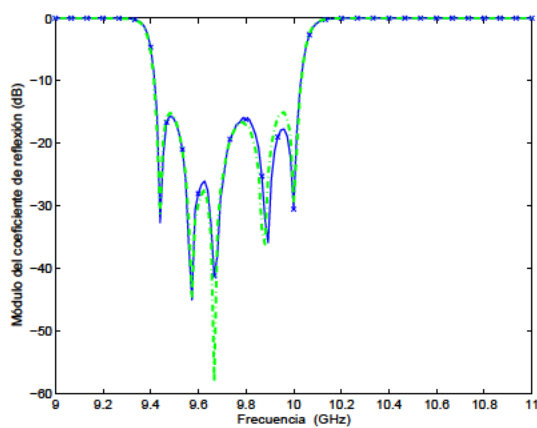


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.38.

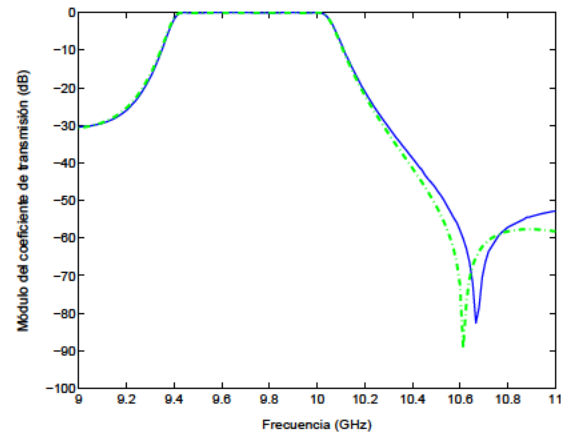
Figura 4.28: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.

obstante, la precisión es similar a la obtenida con la mejor versión de las RBFNNs. Según se aprecia en la Figura 4.30, en la segunda cavidad, el nuevo método también alcanza una elevada precisión en la consecución de la respuesta del filtro de tipo horquilla dibujado en la Figura 3.69. En este caso, la localización del cero de transmisión es mucho más precisa que en el análisis realizado con RBFNNs. Por otra parte, los dos métodos consiguen resultados similares en el parámetro de reflexión S_{11} .

El nuevo método neuronal consigue los resultados mostrados en las Figuras 4.27 y 4.30 con un número reducido de neuronas. En efecto, en la primera caja el número medio de neuronas en la

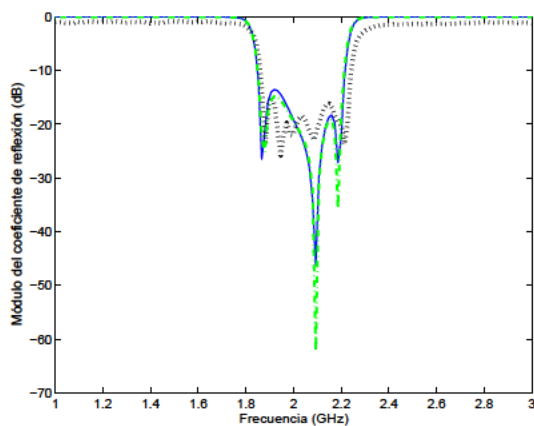


(a) Módulo de S_{11} del filtro paso banda de quinto orden mostrado en la Figura 3.66.

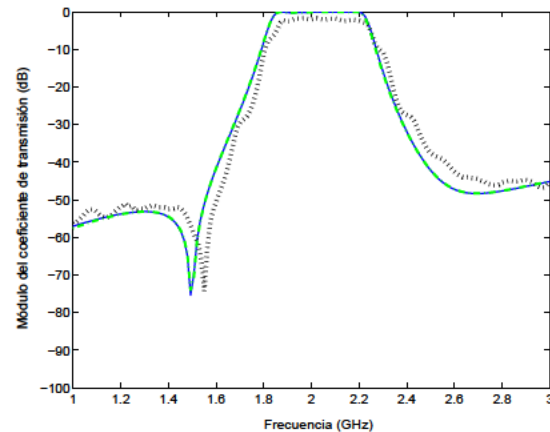


(b) Módulo de S_{21} del filtro paso banda de quinto orden mostrado en la Figura 3.66.

Figura 4.29: Respuesta del filtro de quinto orden de la Figura 3.66 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.69.



(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.69.

Figura 4.30: Respuesta del filtro de quinto orden de tipo horquilla de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.

región MS es de tan solo 88 en la primera cavidad y de 59 en la segunda caja. Las redes neuronales que aproximan las funciones de Green cerca de la fuente son las más demandadas durante la resolución de la IE mediante el MoM debido a que el número de puntos de integración es mayor en esta zona. Gracias a la extracción de dos capas y a la capacidad óptima del algoritmo de entrenamiento de las Redes neuronales de Chebychev, éstas son las más compactas en esta zona. De este modo, en la zona más cercana a la fuente tan solo son necesarias 55 neuronas de media tanto en la primera como en la segunda cavidad. En cambio, en la primera cavidad las RBFNNs combinadas con zonas circulares

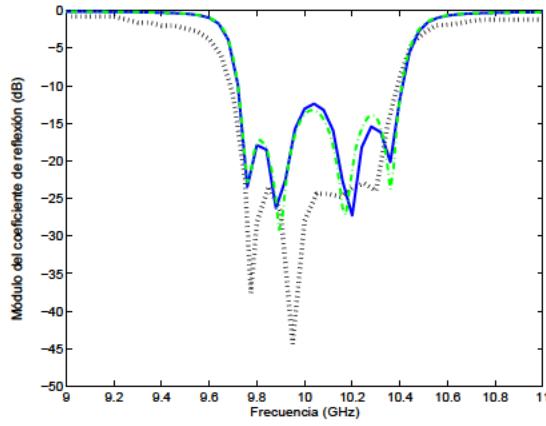
necesitan una media de 226 neuronas en la primera zona y hasta más de 500 neuronas en media para el resto de zonas de la región MS. En la segunda cavidad las RBFNNs combinadas con zonas circulares necesitan una media de 326 neuronas en la primera zona y poco más de 400 neuronas en media para el resto de zonas de la región MS. Además, en la región DS las Redes neuronales de Chebychev poseen menos neuronas que las RBFNNs correspondientes. Así, en la primera cavidad el tamaño neuronal medio en esta región es de 134 y en la segunda cavidad de 139. En cambio, en ambas cavidades el número de neuronas mínimo alcanzado con las RBFNNs se cifra en varios cientos.

De esta manera, la aceleración del análisis de los circuitos encapsulados es notable, como se observa en la Tabla 4.21. El método basado en Redes neuronales de Chebychev es casi ocho veces más rápido que el basado en RBFNNs. El tiempo por punto en frecuencia es de poco más de un segundo para los filtros analizados. El nuevo método es más eficiente que el basado en RBFNNs. La ganancia temporal es mucho más que suficiente para convertir en rentable el tiempo invertido en la selección y entrenamiento de las redes neuronales, aun cuando éste fuera superior al de las RBFNNs.

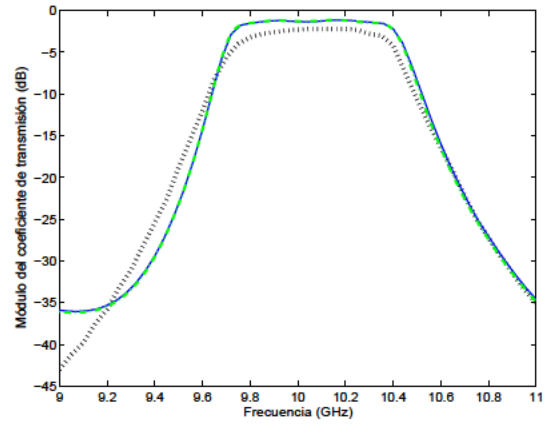
Tabla 4.21: Comparación de los resultados temporales de la solución directa y el método neuronal basado en Redes neuronales de Chebychev para el cálculo de los parámetros S en 151 puntos de frecuencia.

	Solución directa	Neuronal Chebychev	Núm. veces rápido que RBFNNs
Cuatro polos Figura 3.37	112h. 27m.	231 s	7.6
Cuatro polos Figura 3.38	112h. 46m.	238 s	7.6
Cinco polos Figura 3.66	120h. 12m.	253 s	7.8
Cinco polos horquilla Figura 3.69	114h. 18m.	345 s	6.1

Para finalizar la comprobación del método neuronal, las Redes neuronales de Chebychev en las dos cavidades utilizadas han sido entrenadas considerando la existencia de pérdidas en el dieléctrico. De este modo, las redes neuronales deben aproximar las funciones de Green con valores imaginarios distintos de cero. Las zonas observación, fuente y frecuencia en ambas cavidades son las mismas que las empleadas en la aproximación de las funciones de Green en los dieléctricos sin pérdidas. Por tanto, únicamente se debe re-entrenar las redes neuronales utilizadas hasta ahora. La incidencia de los valores imaginarios y el cambio en los valores reales no es suficiente para alterar la división del espacio de entrada. Por ello se puede utilizar la misma división sin que se vea afectada la precisión del método neuronal. En la primera cavidad se ha aplicado una tangente de pérdidas igual a $\tan \delta = 0,0030$. En la segunda cavidad se ha utilizado el valor de pérdidas ($\tan \delta = 0,0020$) que ofrece el fabricante Rogers[®] para su material RO3006 correspondiente al dieléctrico utilizado en esta cavidad. En la Figura 4.31 se dibujan la solución directa, la solución neuronal y las medidas para el filtro de cuarto orden mostrado en la Figura 3.38 impreso en la primera cavidad. Como se observa el método neuronal sigue con precisión las variaciones del parámetro de reflexión y transmisión en todo el ancho de banda. La misma apreciación se obtiene, al observar en la Figura 4.32, la respuesta correspondiente al filtro de tipo horquilla impreso en la segunda cavidad. En este caso las pérdidas son pequeñas y la respuesta es similar a la mostrada en la Figura 4.30.

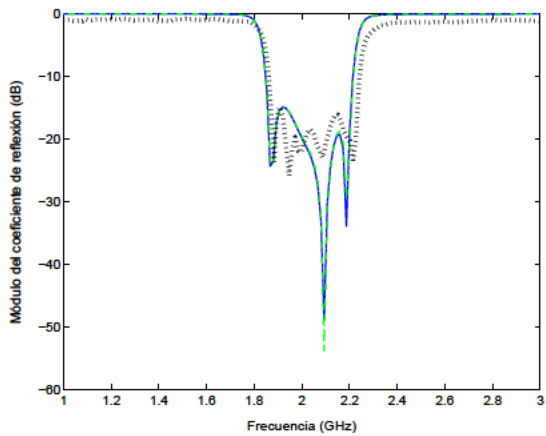


(a) Módulo de S_{11} del filtro paso banda de cuarto orden mostrado en la Figura 3.38 con pérdidas.

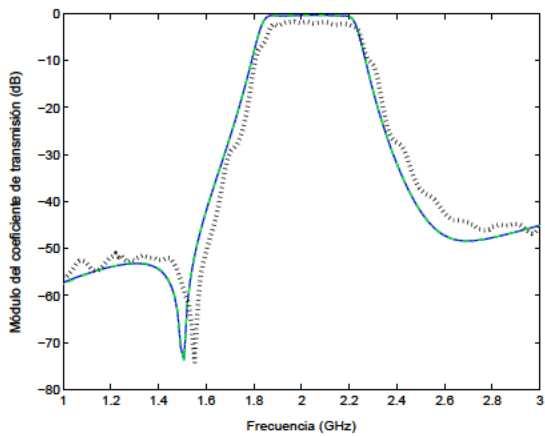


(b) Módulo de S_{21} del filtro paso banda de cuarto orden mostrado en la Figura 3.38 con pérdidas.

Figura 4.31: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas) cuando se consideran pérdidas en el dieléctrico de valor $\tan \delta = 0,003$. Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.



(a) Módulo de S_{11} del filtro paso banda de quinto orden mostrado en la Figura 3.69 con pérdidas.



(b) Módulo de S_{21} del filtro paso banda de quinto orden mostrado en la Figura 3.69 con pérdidas.

Figura 4.32: Respuesta del filtro de quinto orden de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas) cuando se consideran pérdidas en el dieléctrico de valor $\tan \delta = 0,002$. Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.

4.11. Comparativa con otros métodos de análisis

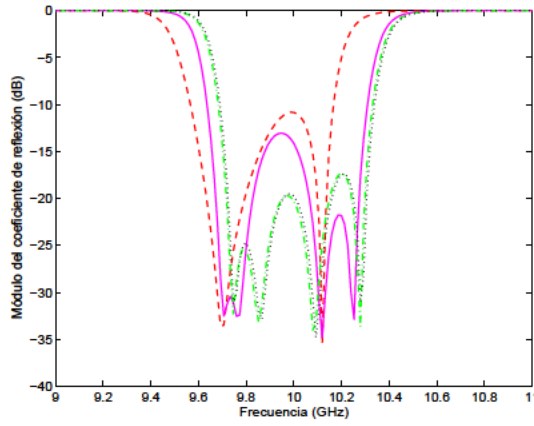
Para completar el estudio sobre el método neuronal de análisis de circuitos multicapa apantallados es necesario comparar su rendimiento con el de otras técnicas y herramientas de análisis. Dicha comparación se va a basar en la rapidez del cálculo de la respuesta que ofrecen los diversos métodos.

La primera de las técnicas utilizadas para realizar la comparación emplea una formulación de las

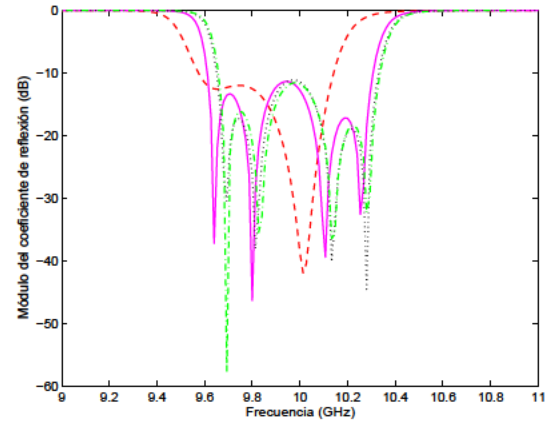
funciones de Green en el dominio espectral [Álvarez Melcón, 1998]. Este tipo de formulaciones son generalmente eficientes, pero en ciertas situaciones la convergencia es lenta y por lo tanto son necesarios un gran número de términos modales para alcanzar la precisión fijada. Este hecho indeseable aparece cuando las dimensiones de la cavidad son grandes en comparación con las dimensiones del circuito impreso. Además, la elección de las funciones base influye de forma decisiva en el rendimiento del método. Por un lado, la convergencia del kernel depende del tipo de funciones base empleado. Por otro lado, el cálculo de las integrales solo se realiza de forma analítica para algunos tipos simples de funciones base [Melcon et al., 1999]. Las formulaciones espaciales son independientes de las dimensiones de la caja y del tipo de funciones de base utilizadas. Por tanto, el análisis del circuito comenzará en las formulaciones espaciales sin ningún estudio previo de convergencia. Como ya se ha comentado en el Capítulo 2 el mayor inconveniente de las formulaciones espaciales reside en la lentitud de la evaluación de las series asociadas. El método neuronal en general, y el basado en Redes neuronales de Chebychev en particular, permite una reducción drástica del tiempo total de análisis respecto al análisis realizado con la formulación espacial de las funciones de Green. Es previsible que el método neuronal será competitivo frente a las formulaciones espectrales, especialmente si la cavidad es mucho mayor que el circuito impreso en su interior.

Para ilustrar este hecho se han analizado los filtros de las Figuras 3.37, 3.38 y 3.66. Estos filtros se encuentran impresos en la cavidad 1 de la Figura 4.18, cuyo tamaño es mucho mayor que las dimensiones de los filtros. En efecto, en las Figuras 4.33 y 4.34 se observa que el método espectral no converge hacia una solución hasta que no se sobrepasan los 75000 modos. Son necesarias diversas pruebas para hallar un número de modos que asegure la estabilidad en la respuesta. Además, el número de modos es muy elevado. Todo ello provoca que el análisis sea lento como se aprecia en la Tabla 4.22. Con este método el tiempo de análisis por punto en frecuencia es cercano al minuto. El método neuronal basado en Redes neuronales de Chebychev en estos filtros es en torno a treinta veces más veloz. Incluso el método neuronal basado en RBFNNs es hasta cuatro veces más rápido que la técnica espectral. La respuesta del parámetro (S_{11}) para la técnica espectral y para el método neuronal se compara en las Figuras 4.35 y 4.36. En todos los casos existe un pequeño desplazamiento de la respuesta del método espectral respecto a la técnica basada en la formulación espacial y el método neuronal. El mallado en todos los filtro es el mismo que se ha utilizado en toda la tesis (por ejemplo, en la Figura 3.41 se muestra el empleado en los filtros de cuarto orden).

Si la cavidad no presenta unas dimensiones mucho mayores que las dimensiones circuitales, entonces la técnica espectral es eficiente. En la Figura 4.37 se aprecia que para alcanzar la convergencia en el filtro de tipo horquilla impreso en la cavidad 2 de la Figura 4.18 son requeridos únicamente 5000 modos. Con este número de modos, la técnica espectral emplea solo 1,30 segundos por punto en frecuencia. En cambio, el método neuronal basado en RBFNNs necesita 13,73 segundos para el mismo filtro con el mismo mallado. Existe por tanto una diferencia apreciable entre ambos métodos, aun sin necesitar el método neuronal en cada circuito un método de estudio previo sobre convergencia. El método neuronal basado en Redes neuronales de Chebychev consigue un tiempo de análisis de tan solo 2,29 segundos por punto en frecuencia. Es un tiempo comparable al alcanzado con la técnica espectral. El nuevo método neuronal es así competitivo con la técnica espectral, incluso en cavidades con un tamaño similar al del circuito impreso. Aunque en el método neuronal es necesario un largo periodo de entrenamiento y selección, éste se debe realizar inicialmente y solo una vez, al contrario



(a) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.37 para la técnica definida en el dominio espectral. Diversos números de modos.



(b) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.38 para la técnica definida en el dominio espectral. Diversos números de modos.

Figura 4.33: Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de las Figuras 3.37 y 3.38 para la técnica definida en el dominio espectral. La respuesta se muestra para 25000 modos (línea con guiones), 50000 modos (línea continua), 75000 modos (línea con guión y puntos) y para 100000 modos (línea punteada).

que el estudio de convergencia en el método espectral. Este estudio es aconsejable realizarlo antes del análisis de cada tipo de filtro. La comparación entre las respuestas de ambos métodos en el filtro de tipo horquilla se encuentra dibujada en la Figura 4.38. En este caso, las respuestas obtenidas con la formulación espacial, con el método neuronal de Chebychev y con la técnica espectral son similares en todo el ancho de banda.

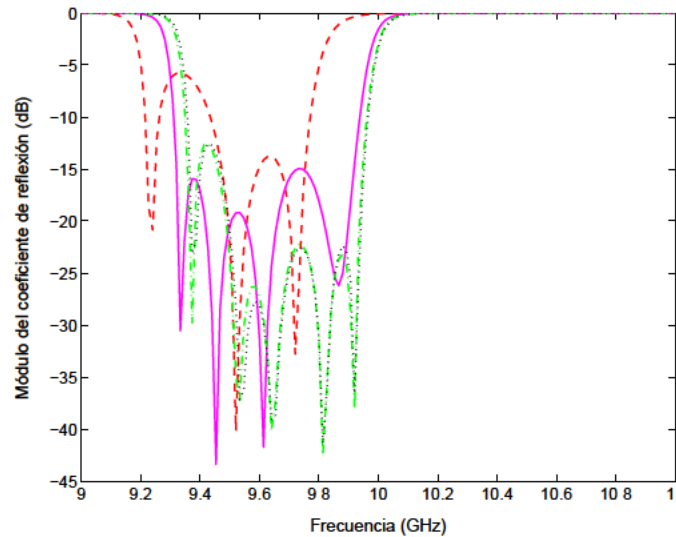
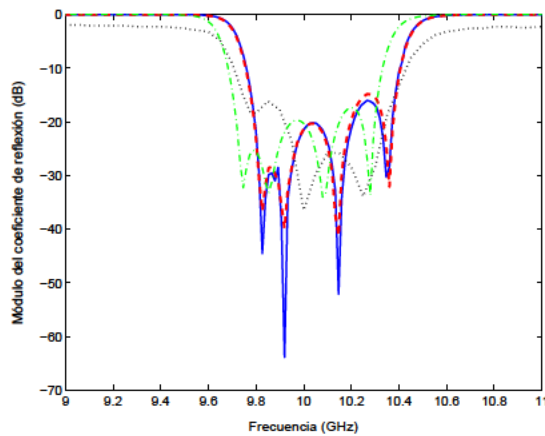
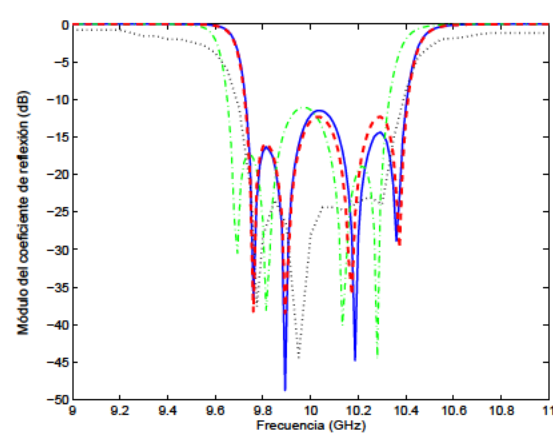


Figura 4.34: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espectral. La respuesta se muestra para 25000 modos (línea con guiones), 50000 modos (línea continua), 75000 modos (línea con guión y puntos) y para 100000 modos (línea punteada).



(a) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.37.



(b) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.38.

Figura 4.35: Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de las Figuras 3.37 y 3.38 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 75000 modos (línea con guión y puntos) y el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).

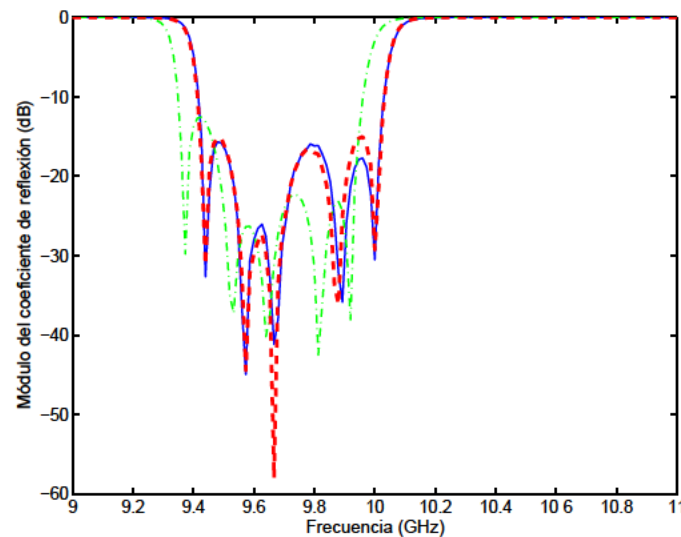


Figura 4.36: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 75000 modos (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones).

Los filtros encapsulados también han sido analizados mediante el programa comercial ADS[®] en su versión del año 2006. Este programa utiliza la técnica de la IE resuelta con el MoM y es capaz de analizar tanto estructuras encapsuladas como estructuras sin apantallamiento. Las respuestas obtenidas para los filtros impresos en la cavidad primera se muestran en las Figuras 4.39 y 4.40 mientras que la respuesta del filtro de tipo horquilla de la segunda cavidad se observa en la Figura 4.41. En los filtros de líneas acopladas se ha empleado un mallado con tres filas de celdas de igual ancho

Tabla 4.22: Comparación de los resultados temporales del método neuronal basado en Redes neuronales de Chebychev, de la técnica de análisis definida en el dominio espectral (75000 modos) y del programa comercial ADS[®]. Todos los resultados corresponden al tiempo por punto en frecuencia en segundos.

	Neuronal Chebychev	Método espectral	ADS [®]
Cuatro polos Figura 3.37	1.53	46.39	5.38
Cuatro polos Figura 3.38	1.58	46.42	5.64
Cinco polos Figura 3.66	1.67	51.83	5.56
Cinco polos horquilla Figura 3.69	2.29	1.30	9.00

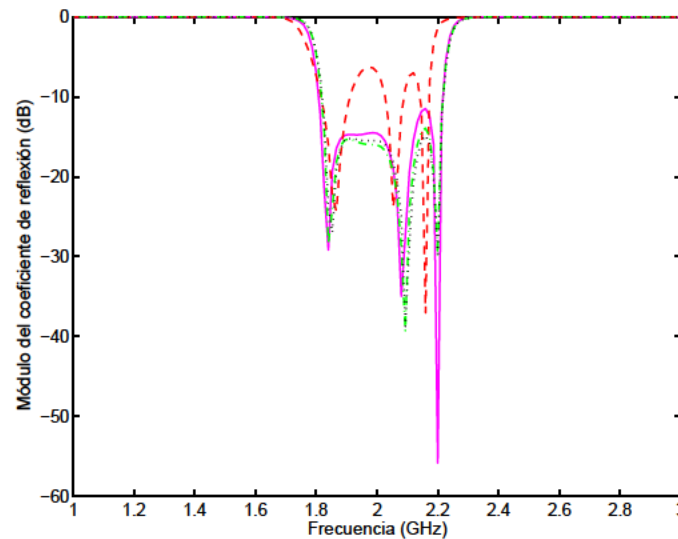


Figura 4.37: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espectral. La respuesta se muestra para 1500 modos (línea con guiones), 2500 modos (línea continua), 5000 modos (línea con guión y puntos) y para 10000 modos (línea punteada).

como en los mallados utilizados en el método neuronal basado en RBFNNs y en el basado en Redes neuronales de Chebychev. El número de columnas se ha ajustado, tanto en el caso de los filtros de líneas acopladas como en el filtro de tipo horquilla, para que el número de celdas total en el simulador ADS[®] y los métodos neuronales sea similar. De este modo la comparación temporal es más certera. El tiempo de simulación de todos los filtros se enumera en la Tabla 4.22. Como puede apreciarse el método neuronal basado en Redes neuronales de Chebychev es siempre más rápido que el programa comercial. El método neuronal basado en RBFNNs es en cambio más lento que la versión de ADS[®] empleada. Aunque la diferencia temporal entre el método neuronal de Chebychev y ADS[®] es menor que la obtenida con el método espectral en la primera cavidad, el método neuronal resulta rentable conforme aumentamos el número de análisis. Si en una cavidad se van a diseñar diversos filtros, será

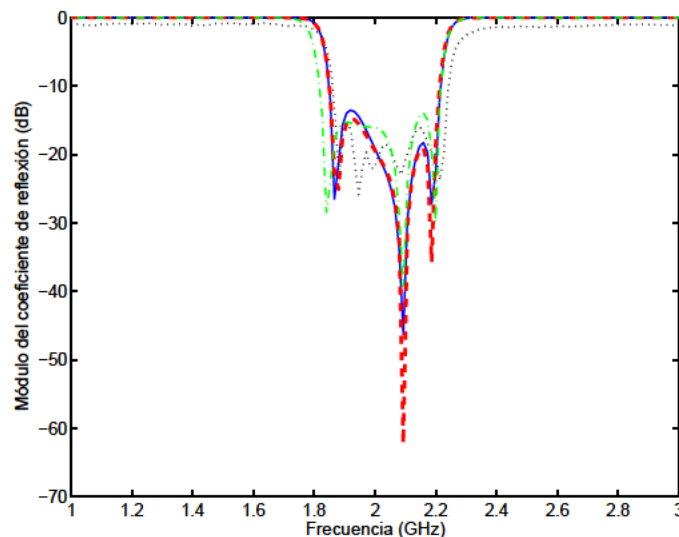
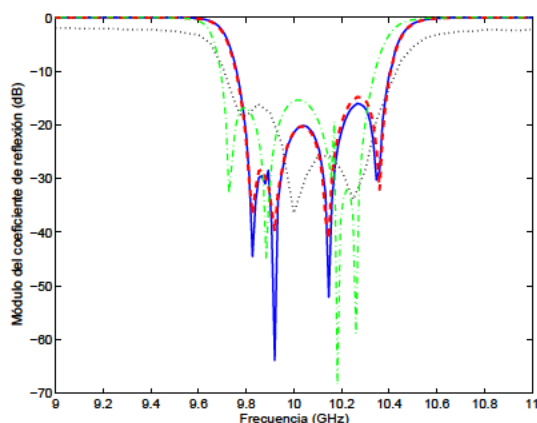
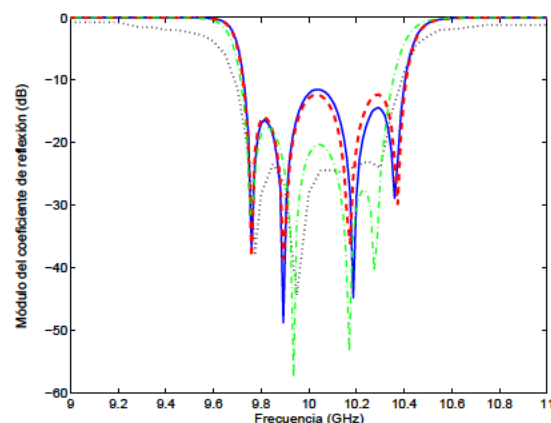


Figura 4.38: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 5000 modos (línea con guión y puntos) y el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).



(a) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.37.



(b) Módulo del coeficiente de reflexión S_{11} del filtro de cuarto orden de la Figura 3.38.

Figura 4.39: Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de la Figuras 3.37 y 3.38 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS^{\circledast} (línea con guión y puntos), y para el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).

necesario realizar decenas de simulaciones en cada uno de ellos, por lo que el tiempo invertido en la selección y entrenamiento de las redes neuronales llegará a ser despreciable frente a la ganancia temporal conseguida.

Se puede argüir que una mejora en la evaluación de las funciones de Green de la formulación espacial acabaría con la rentabilidad del método neuronal. Si la evaluación de las funciones de Green

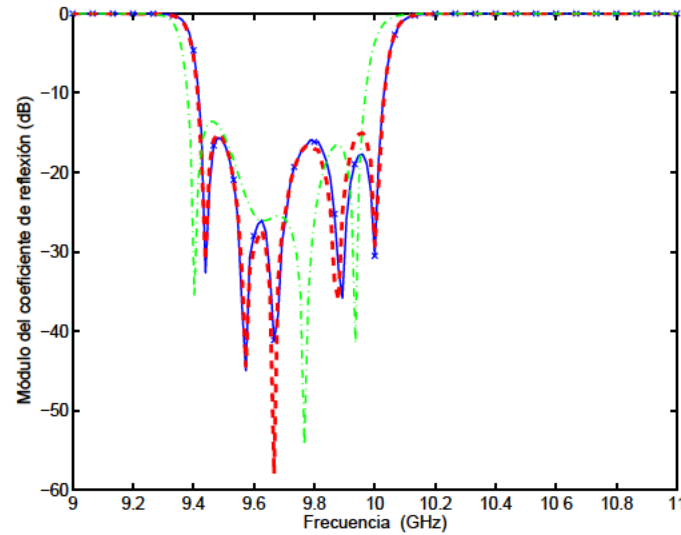


Figura 4.40: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS[®] (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones).

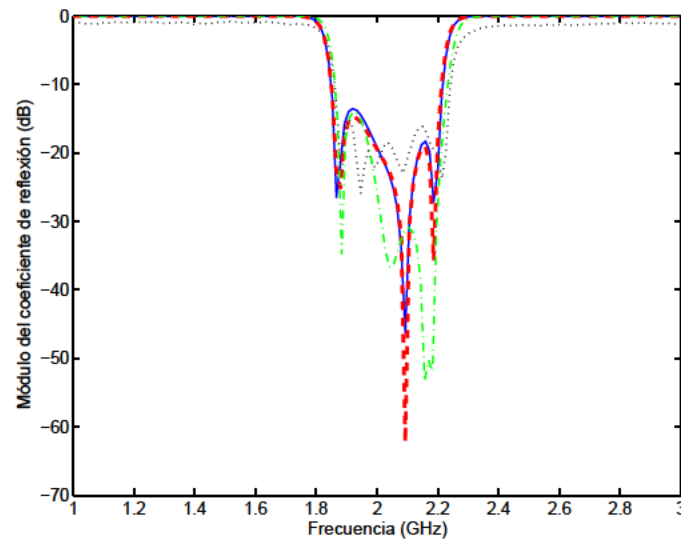


Figura 4.41: Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS[®] (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).

fuera mucho más rápida, el método neuronal perdería eficiencia al reducirse la ganancia temporal. Pero en este caso, el tiempo de división del espacio de entrada así como el tiempo de entrenamiento se vería reducido al disminuir el tiempo de generación de los conjuntos de entrenamiento. Hay que tener en cuenta que, para los tamaños de los conjuntos de entrenamiento utilizados, el tiempo de entrenamiento es mucho menor que el tiempo de generación de dichos conjuntos. Por ejemplo, en la Tabla 4.10 se observa que el tiempo empleado para seleccionar 500 regresores de un conjunto de

entrenamiento de 12005 muestras es de poco más de un minuto. Para este tamaño de entrenamiento el tiempo empleado en la generación de los conjuntos de entrenamiento y validación es de 401 segundos. Para un tamaño de entrenamiento de 2401 muestras el tiempo de entrenamiento dura en torno a 3 segundos, mientras que el periodo de generación es hasta 21 veces mayor.

Ante una mejora de la evaluación de las funciones de Green, existirá una cierta cantidad de análisis circuitales que convertirá en rentable el método neuronal. Consideraremos que el método es rentable cuando el tiempo ahorrado en el análisis de los circuitos sea igual al tiempo empleado en la producción de las redes neuronales. Para estimar la cantidad de análisis circuitales necesarios para alcanzar la rentabilidad se han estudiado diferentes casos. En cada caso se supone el análisis de un circuito que conlleva un número determinado de evaluaciones de las funciones de Green en la solución directa y de evaluaciones de redes neuronales en el método neuronal. Asimismo, en cada caso se estudian hasta cuatro velocidades diferentes de evaluación de las funciones de Green. Dependiendo de la cercanía a la fuente la rapidez en el cálculo de las funciones de Green será mayor o menor. En este estudio, el tiempo de evaluación es el tiempo medio dedicado a esta tarea en el análisis de un circuito. Es decir, es el fruto de dividir el tiempo total dedicado a la evaluación de las funciones de Green en el análisis de un circuito por el número de evaluaciones. No se distingue entre aquellas evaluaciones en la región DS y en la región MS; ambos tiempos se encuentran mezclados. Respecto al tiempo de generación de las Redes neuronales de Chebychev se ha considerado que el tiempo de entrenamiento es el 50 % del total. Este es un valor conservador; tanto en la primera como en la segunda cavidad de la Figura 4.18 el tiempo real se aproxima más a un valor del 40 %. Asimismo, hemos sido muy conservadores en la fijación del tiempo de división del espacio de entrada. En todos los casos se supone que es igual a 30 horas; además no se incluye en este periodo el ahorro de tiempo debido a la mejora en la evaluación de las funciones de Green. El estudio se ha realizado teniendo en cuenta las redes neuronales y densidades de mallado utilizados en el análisis de los filtros de las cavidades de la Figura 4.18. Así, el número de evaluaciones para entrenar todas las redes neuronales en la primera cavidad es de 7002207. En la segunda cavidad el número asciende a 9997759. Los mallados estudiados son tres en ambas cavidades. En la primera cavidad, el mallado utilizado en los filtros de cuatro polos requiere de 200512 ejecuciones de las funciones de Green con una rapidez de 0,013 segundos para cada par de puntos fuente y observación. Por ello, se ha usado un número de ejecuciones de 200000; un número de 166000 correspondiente a un mallado menos denso y un total de 397000 ejecuciones correspondiente a un mallado más denso. En la segunda cavidad, el mallado empleado en el filtro de tipo horquilla implica 225320 evaluaciones de las funciones de Green. Siguiendo la misma filosofía que en la primera cavidad, se ha utilizado un valor de 225000, 140000 y 420000 evaluaciones de Green correspondientes a tres grados de mallado.

En las Figuras 4.42 y 4.43 se muestran el número de análisis circuitales necesarios en la primera y segunda cavidad respectivamente para que el método neuronal sea competitivo. Se han considerado en cada gráfica cuatro velocidades de evaluación de las funciones de Green: 10^{-1} , 10^{-2} , 10^{-3} y 10^{-4} segundos por pareja de puntos fuente y observación. Para ambas cavidades se toman los mismos tiempos ya que se ha comprobado que el tiempo medio habitual era similar. En el tiempo de análisis se incluye el periodo empleado para resolver el sistema matricial obtenido en el MoM. En todas las gráficas el eje x indica el número de veces que el método neuronal es más rápido que el método empleado para evaluar numéricamente las funciones de Green. Cada análisis la respuesta se calcula

para 100 puntos en frecuencia. En las dos Figuras 4.42 y 4.43 cuando el número de análisis es menor que uno, significa que con tan solo un análisis el método neuronal sería rentable. Como se aprecia en las gráficas para los valores habituales empleados en el análisis de los filtros de líneas acopladas y de tipo horquilla (rapidez de 10^{-2} segundos y mallado intermedio) son necesarios muy pocos análisis circuitales. Hay que tener en cuenta que el método neuronal de Chebychev es casi tres órdenes de magnitud más rápido que el método directo. Conforme el método de evaluación numérica es más veloz se requiere un número mayor de análisis. En el caso de 10^{-4} segundos es apropiado que la red neuronal sea hasta 10 veces más rápida. Este es un valor factible ya que actualmente, con el método de Chebychev y en término medio, el tiempo requerido para obtener las salidas neuronales es de aproximadamente 10^{-5} segundos. De este modo, con menos de una centena de análisis neuronales, el método sería rentable. Esta situación es plausible si se diseñan unos pocos circuitos en una misma cavidad. Para velocidades de evaluación directa menores de las mostradas, las redes neuronales deberían ser reducidas en su tamaño para acelerar el análisis mediante el método neuronal. Obviamente, siempre existe un número de análisis circuitales que permiten que el método neuronal sea rentable ya que consideramos que la red neuronal es siempre, en término medio, más rápida que la evaluación directa. Este hecho es perfectamente factible debido a que la evaluación implica la suma de series de difícil convergencia mientras que el cálculo neuronal es muy sencillo. Además, el cálculo de las salidas neuronales es fácilmente paralelizable. Esta característica posibilita el uso de FPGAs ("Field Programmable Gate Array") que aceleren de forma sustancial el cálculo de las salidas neuronales. En todos los casos se observa que el número de análisis tiende a un valor constante. Esto es debido a que por mucho más rápida que sea la red neuronal respecto a la evaluación numérica de las funciones de Green, es decir, por mucho más rápido que sea el llenado de la matriz de impedancias generalizadas, siempre tenemos un tiempo mínimo de obtención de la respuesta correspondiente al tiempo de resolución de los sistemas lineales. Todas las simulaciones de esta Sección se realizaron con un ordenador con un procesador Pentium IV a 3GHz y con 4GB de memoria RAM.

4.12. Análisis de circuitos encapsulados con Redes neuronales de Chebychev en cavidades variables

Las Redes neuronales de Chebychev carecen de la limitación de las RBFNNs en el tamaño del conjunto de entrenamiento cuando ambas son entrenadas con el algoritmo OLS. Por lo tanto, las nuevas redes pueden aproximar las funciones de Green en espacios de entrada de dimensión superior a cinco. Estos espacios de entrada requieren conjuntos de entrenamiento de decenas de miles de muestras. Por ejemplo, una zona fuente con un valor típico de 45 puntos fuente, para una zona observación muestreada en 35 puntos, en una zona frecuencial muestreada en tan solo 3 puntos genera un conjunto de entrenamiento de 4725 puntos. Si se desean incluir en el método neuronal las variaciones debidas a las dimensiones transversales de la cavidad, entonces hay que añadir dos nuevas dimensiones en el espacio de entrada. Un muestreo de 3 puntos en estas dos dimensiones produce un nuevo conjunto de entrenamiento de 51975 puntos. Si el número de puntos frecuencia se eleva a 5, el tamaño aumenta hasta las 86625 muestras. El entrenamiento de estos conjuntos con una RBFNN se torna en imposible debido a las limitaciones de memoria. Con las Redes neuronales de Chebychev la aproximación es posible. Sin embargo, para que el método neuronal sea práctico hay que superar dos inconvenientes. El tiempo de entrenamiento va a aumentar en gran medida, no tanto por la selección

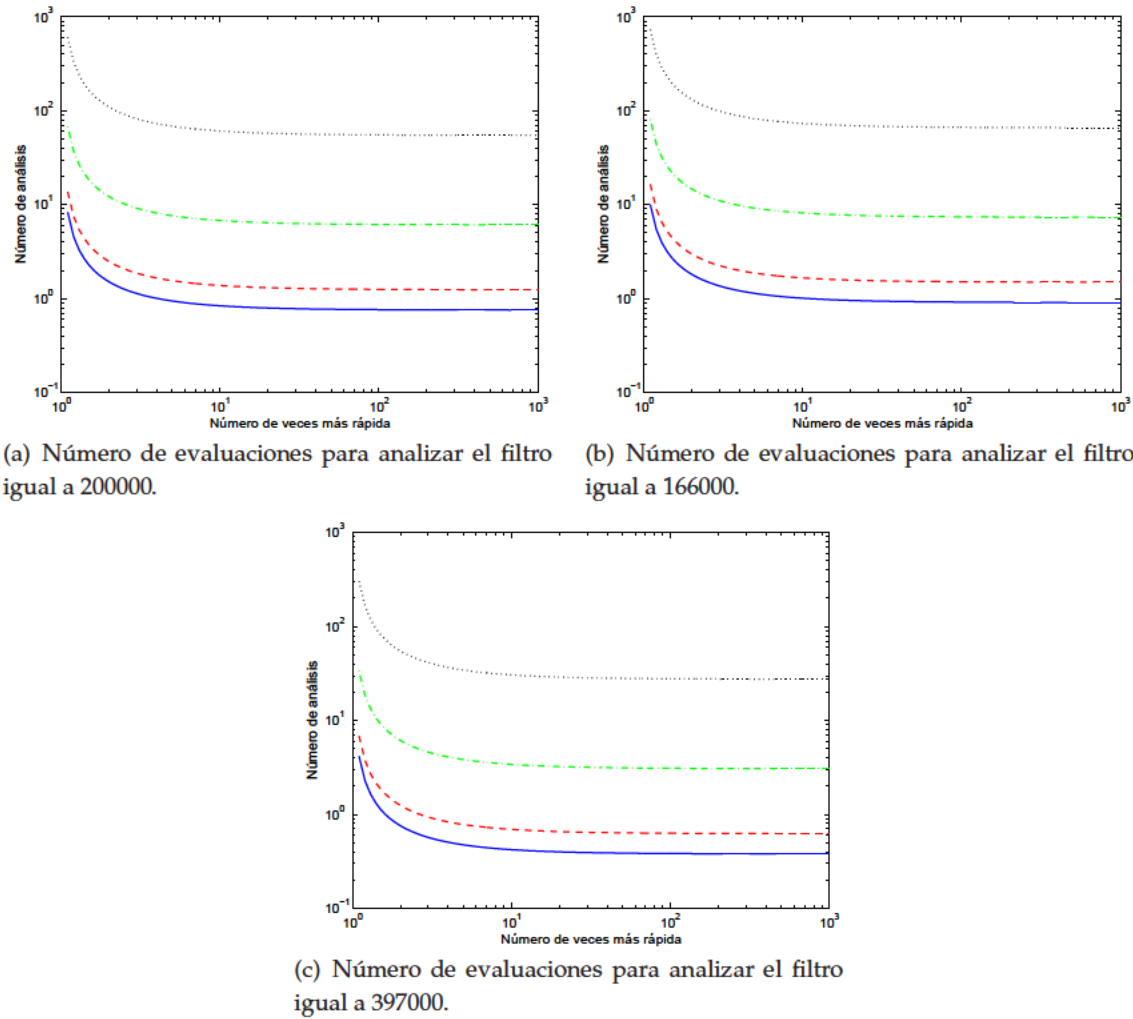


Figura 4.42: Número de análisis circuitales necesarios para convertir en rentable el método neuronal de Chebychev en un filtro típico impreso en la primera cavidad de la Figura 4.18. Se muestran los resultados para un valor medio de evaluación de las funciones de Green de 10^{-1} segundos (línea continua), 10^{-2} segundos (línea con guiones), 10^{-3} segundos (línea con guión y puntos), 10^{-4} segundos (línea punteada).

de los mejores regresores, sino por el tiempo requerido para generar los conjuntos de entrenamiento. Además, los tamaños neuronales deben ser reducidos para que la ganancia temporal sea elevada y convierta en despreciable el tiempo de entrenamiento. Para comprobar la viabilidad del método neuronal en espacios de entrada mayores se han realizado dos pruebas en dos cavidades distintas.

La primera cavidad utilizada es aquella marcada con un 1 en la Figura 4.18. Las funciones de Green en esta cavidad son aproximadas para variaciones del valor de permitividad y el espesor de la capa dieléctrica. De esta forma, cuando se analice un circuito con el método neuronal, el valor de ϵ_{r2} podrá tomar un valor entre 9,00 y 10,80. El espesor del dieléctrico podrá tomar valores entre 0,35 mm y 0,45 mm. Así, los circuitos ya analizados en esta cavidad podrán volverse a estudiar incorporando las derivas en el valor de permitividad o nuevos valores de espesor del dieléctrico. Las zonas fuente, observación y frecuencia son las mismas que las mostradas en la Sección 4.10. En cada

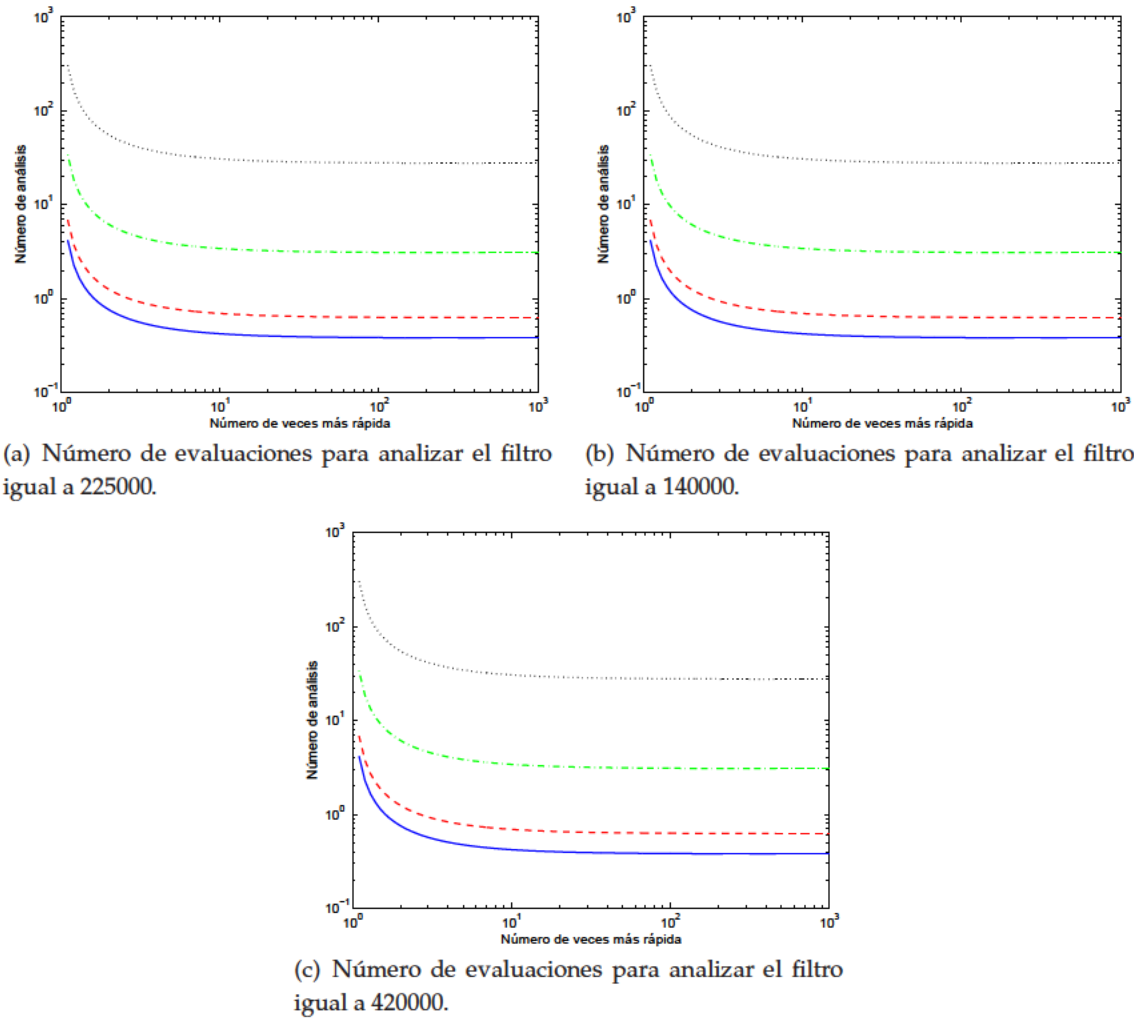


Figura 4.43: Número de análisis circuitales necesarios para convertir en rentable el método neuronal de Chebychev en un filtro típico impreso en la segunda cavidad de la Figura 4.18. Se muestran los resultados para un valor medio de evaluación de las funciones de Green de 10^{-1} segundos (línea continua), 10^{-2} segundos (línea con guiones), 10^{-3} segundos (línea con guión y puntos), 10^{-4} segundos (línea punteada)

una de las nuevas dimensiones del espacio de entrada se genera una única zona muestreada en 3 puntos. Se asume que la división de las coordenadas espaciales y de la variable frecuencia es válida para cualquier valor de permitividad y espesor del dieléctrico. Así, la cavidad 1 es una estructura que sirve para dividir parte del espacio de entrada. Esta división se extiende para cualquier nueva cavidad formada con valores de permitividad y espesor dentro de los límites establecidos.

El muestreo realizado produce conjuntos de entrenamiento con tamaños entre las 25000 y las 55000 muestras. Del mismo modo que en el método neuronal explicado en las secciones precedentes, el espacio de entrada es muestreado de forma más densa para formar el conjunto de validación que para formar el conjunto de entrenamiento. Por ello, los conjuntos de validación poseen tamaños entre 62000 y 110000 puntos. De esta forma, se comprueba de modo exigente la capacidad de generalización de las Redes neuronales de Chebychev, especialmente en la frontera entre zonas, donde no

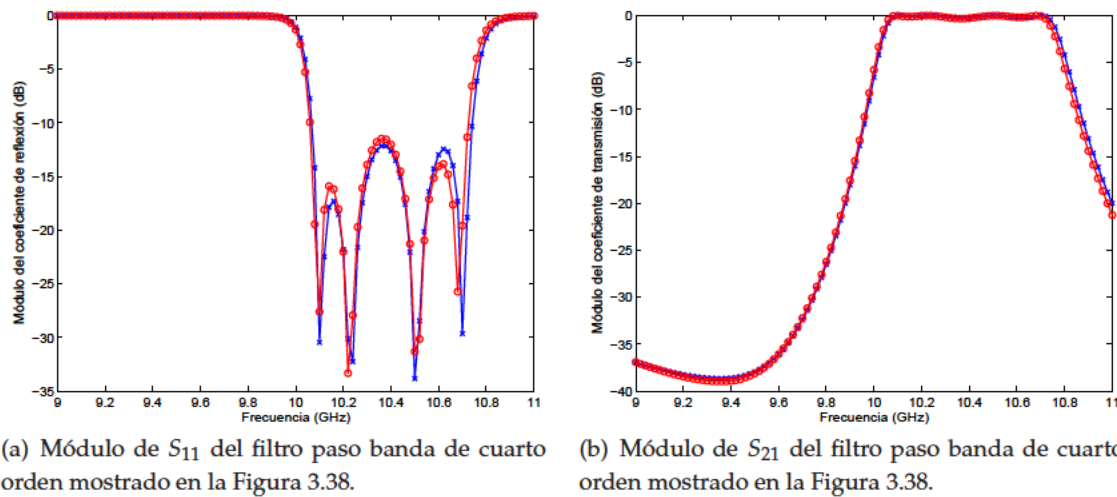


Figura 4.44: Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 101 puntos de frecuencia.

existen puntos de entrenamiento. El número máximo de neuronas para cada Red neuronal de Chebychev se fijó en 250. Con este tamaño máximo neuronal y para un error NMSE máximo de 0,025, el tiempo de entrenamiento de todas las redes neuronales fue de 126 horas y 13 minutos, muy superior por tanto al tiempo necesario cuando el espacio de entrada es de cinco dimensiones. El exhaustivo entrenamiento posibilita que el método neuronal ampliado sea preciso en el análisis de los filtros estudiados previamente. En la Figura 4.44 se muestra la respuesta exacta y la solución neuronal para el filtro de cuarto orden de la Figura 3.38 cuando la permitividad tiene un valor de 9,20 y el espesor del dieléctrico es de 0,381 mm. Estos valores corresponden al material TMM10 del fabricante *Rogers*®. De este modo se puede estudiar el comportamiento del filtro si en lugar de utilizar el material inicialmente escogido se emplea otro material.

El método neuronal ampliado a espacios de entrada de dimensión 7 ha sido empleado en una nueva cavidad, distinta de las usadas hasta el momento. En este caso se permite la variación de la anchura (b) y longitud (a) de la cavidad dibujada en las Figuras 4.45 y 4.48. La permitividad y grosor de la capa dieléctrica permanecen constantes. La longitud tomará valores entre 20 mm y 50 mm. La anchura podrá variar entre 10 mm y 40 mm. Al igual que en el ejemplo anterior, es necesario definir las zonas observación, fuente y frecuencia en una cavidad base. Se ha elegido una cavidad en la que la longitud y anchura son los valores medios del rango de variación de cada dimensión. Por lo tanto, la longitud en la cavidad base es de 35 mm y la anchura tiene un valor de 25 mm. Siguiendo el procedimiento explicado en la Sección 2.4, el umbral $U_x = U_y$ que delimita la región MS y DS se fijó en un valor de $0,10/\pi$. En las Tablas 4.23 y 4.24 se muestran los límites de las zonas en la región MS y la región DS respectivamente. La región DS es mucho mayor que la región cercana a la fuente de modo que son necesarias hasta ocho zonas para dividir completamente la región de débil singularidad. El número de zonas fuente es expuesto en las Tablas 4.25, 4.26 y 4.27. Los límites de cada una de las zonas fuente son mostrados en el Apéndice D. Para calcular las zonas observación se invirtió una 1 hora y 24 minutos. La división de los puntos fuente requirió de 9 horas y 20 minutos. En el ancho de banda de trabajo situado entre 1 GHz y 3 GHz existe una única zona, muestreada en

3 puntos. El grado de división es similar al del resto de cavidades utilizadas. De la misma manera que la componente frecuencial, las dos nuevas dimensiones no son divididas en múltiples zonas. El rango de variación delimita una única zona en cada caso que debe ser muestreada con 4 puntos. Este muestreo posibilita un error NMSE máximo de 0,025 en la región MS y de 0,035 en la región DS. La aproximación en el rango de variación de las dimensiones de la cavidad es difícil y por ello el muestreo es mayor que en la componente frecuencial. Por ello, los conjuntos de entrenamiento alcanzan tamaños entre 66000 y 92000 muestras, mayores que los necesitados con la cavidad anterior. Los conjuntos de validación superan las 100000 muestras en todos los casos, hasta un valor máximo de 147000 puntos. El tamaño máximo neuronal se fijó en 350 neuronas. Con estos valores, el tiempo total de generación de las redes neuronales para todas las combinaciones de zonas fue de 340 horas.

Tabla 4.23: Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,10/\pi$ en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$). Técnica de eliminación de la singularidad mediante la extracción de dos capas.

Zona	Límite inferior	Límite superior
1ª zona	Posición del punto fuente	$0,0500/\pi$
2ª zona	$0,0500/\pi$	$0,1000/\pi$

Tabla 4.24: Zonas observación generadas en la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$). Técnica de eliminación de la singularidad mediante la multiplicación por (ρ).

Zona	Límite inferior	Límite superior
1ª zona	$0,100/\pi$	$0,200/\pi$
2ª zona	$0,200/\pi$	$0,350/\pi$
3ª zona	$0,350/\pi$	$0,500/\pi$
4ª zona	$0,500/\pi$	$0,575/\pi$
5ª zona	$0,575/\pi$	$0,637/\pi$
6ª zona	$0,637/\pi$	$0,700/\pi$
7ª zona	$0,700/\pi$	$0,850/\pi$
8ª zona	$0,850/\pi$	1,000

Tabla 4.25: Número de zonas fuente correspondiente a las zonas observación de la región MS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).

Tipo de zona	1ª zona observ.	2ª zona observ.
Primera zona	4	—
Arriba	—	4
Abajo	—	5
Derecha	—	4
Izquierda	—	5

Tabla 4.26: Número de zonas fuente correspondiente a las cuatro primeras zonas observación de la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).

Tipo de zona	1ª zona observ.	2ª zona observ.	3ª zona observ.	4ª zona observ.
Arriba	4	4	4	4
Abajo	5	5	4	4
Derecha	4	4	4	4
Izquierda	5	4	4	4

Tabla 4.27: Número de zonas fuente correspondiente a las cuatro últimas zonas observación de la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).

Tipo de zona	5ª zona observ.	6ª zona observ.	7ª zona observ.	8ª zona observ.
Arriba	4	4	4	4
Abajo	3	4	3	3
Derecha	4	4	4	4
Izquierda	3	3	3	3

Una vez entrenadas todas las redes neuronales, se pueden analizar rápidamente circuitos impresos en cavidades con diferentes dimensiones transversales dentro de los rangos establecidos. En la Figura 4.45 se muestra un filtro de segundo orden formado con resonadores de tipo anillo cuadrado abierto en una cavidad con dimensiones $a \times b = 29,83mm \times 20,00mm$. Su mallado está dibujado en la Figura 4.46. La solución directa y la solución neuronal se muestran en la Figura 4.47. Aunque existe un pequeño desplazamiento en frecuencia, la precisión es adecuada.

Un segundo filtro ha sido analizado en una cavidad diferente de la anterior, con dimensiones transversales $a \times b = 48,78mm \times 34,00mm$ como se aprecia en la Figura 4.48. El filtro es de segundo orden y está formado por resonadores de tipo anillo cuadrado cerrado. En este caso, la dimensión longitud de la cavidad se encuentra muy cerca de uno de los extremos de la zona correspondiente de modo que la habilidad de las Redes neuronales de Chebychev es puesta a prueba de nuevo de forma exigente. La respuesta correspondiente al método neuronal y a la solución directa se muestran en la Figura 4.50. Como se aprecia, el método neuronal sigue con fidelidad la evolución de los parámetros de reflexión y transmisión en el ancho de banda seleccionado.

En esta sección se ha demostrado que el método neuronal de Chebychev es capaz de aproximar las funciones de Green cuando se permite la variación de ciertas características de la estructura. Sin embargo, el tiempo de entrenamiento es muy elevado de modo que el método neuronal no es especialmente práctico. La aproximación es más complicada, de forma que el número máximo de neuronas y también el número de neuronas medio aumenta. Por tanto, la aceleración del análisis circuital es menor que en el caso de que el espacio de entrada posea cinco dimensiones. Todo ello, especialmente el elevado tiempo de entrenamiento, contribuye a que el método neuronal únicamente sea una herramienta competitiva si se utiliza en multitud de ocasiones. Este hecho puede suceder si

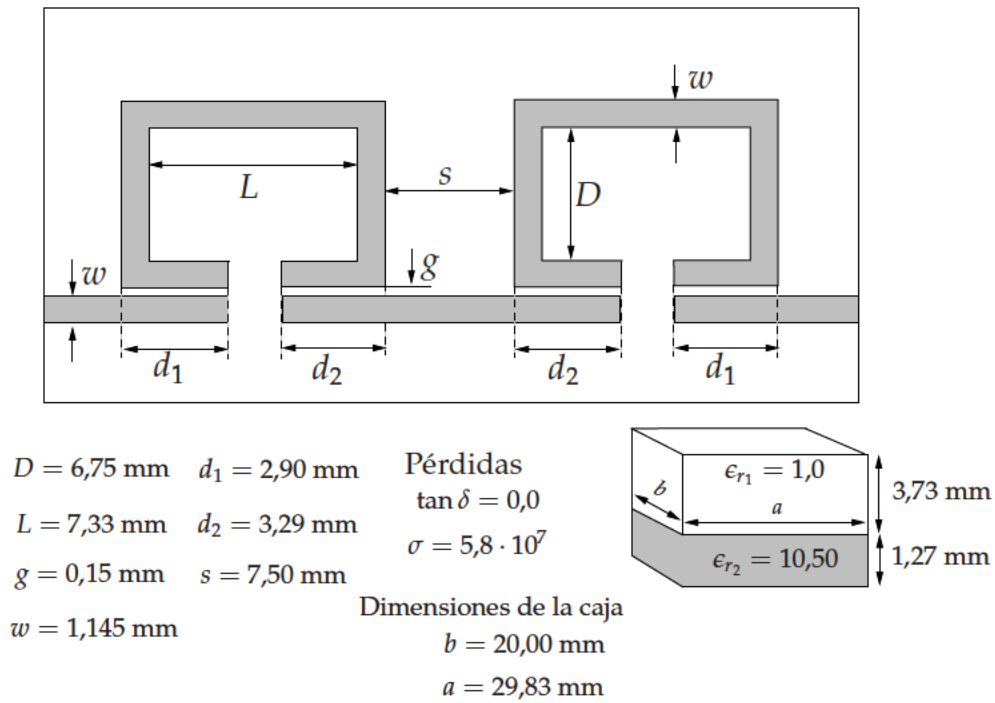


Figura 4.45: Filtro de segundo orden con anillos cuadrados abiertos. Se incluye la estructura del filtro y las dimensiones de la caja.

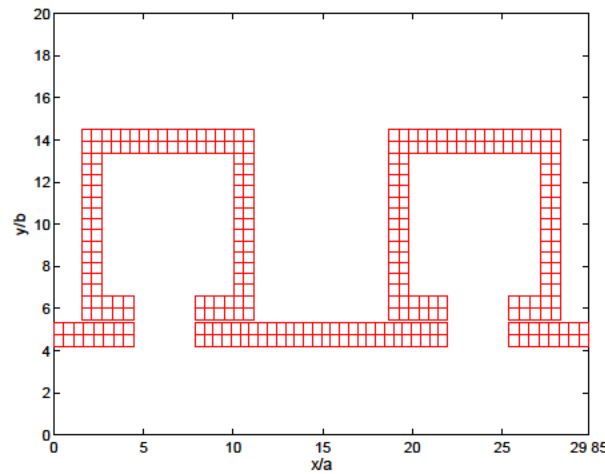
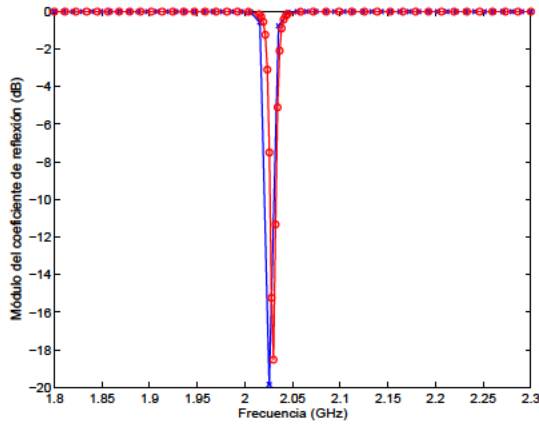


Figura 4.46: Mallado del filtro con anillos cuadrados abiertos.

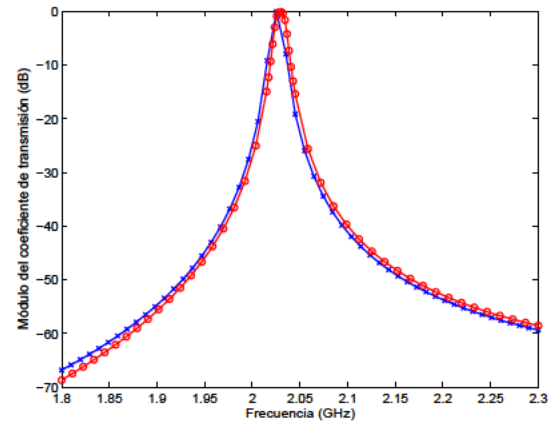
se diseñan varios filtros en distintas estructuras.

4.13. Conclusiones

En este Capítulo se ha aplicado de forma novedosa el algoritmo de entrenamiento OLS en una red neuronal de tres capas formada por polinomios de Chebychev. Hasta el momento nunca se había aplicado el algoritmo OLS en ninguna de las estructuras neuronales de Chebychev publicadas en la literatura científica. Gracias a la propiedad de ortogonalidad en los polinomios de Chebychev se



(a) Módulo de S_{11} del filtro paso banda de segundo orden mostrado en la Figura 4.45.



(b) Módulo de S_{21} del filtro paso banda de segundo orden mostrado en la Figura 4.45.

Figura 4.47: Respuesta del filtro de segundo orden de la Figura 4.45 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 52 puntos de frecuencia.

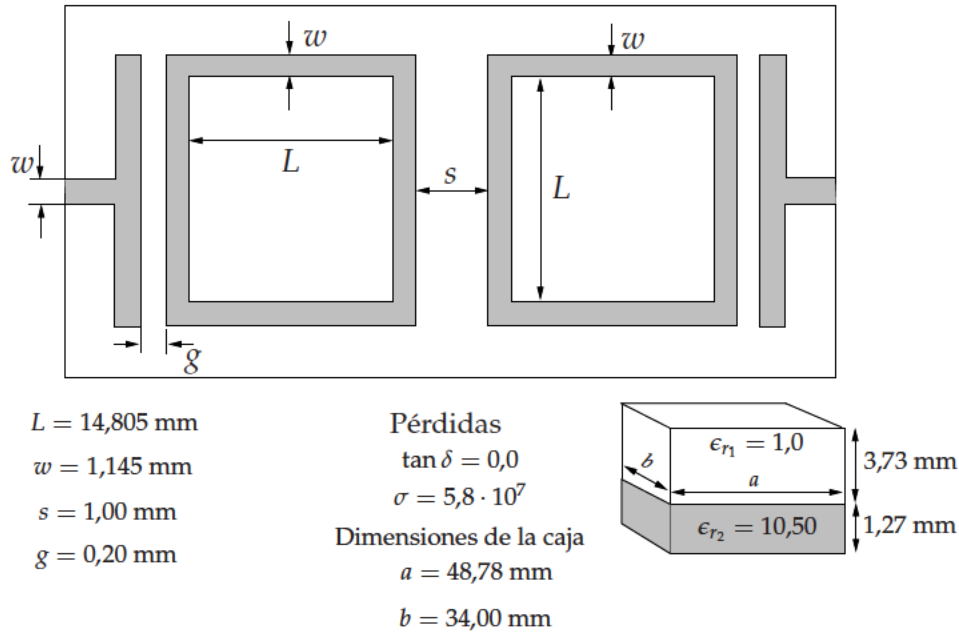


Figura 4.48: Filtro de segundo orden con anillos cuadrados cerrados. Se incluye la estructura del filtro y las dimensiones de la caja.

pueden generar una serie de regresores inicialmente ortogonales que forman el conjunto de entrenamiento. Cuando se da esta circunstancia el algoritmo OLS alcanza un rendimiento óptimo. Por ello, el número de regresores o neuronas seleccionado es mínimo respecto al criterio de expresión de la energía de salida utilizado en el algoritmo OLS. Además, el cálculo de dichos regresores es muy simple, de forma que el tiempo de selección es asimismo minimizado. Si el espacio de entrada es unidimensional las neuronas son simplemente polinomios de Chebychev de modo que la propiedad de ortogonalidad es cumplida fácilmente. En este caso, el conjunto de entrenamiento se obtiene mediante el muestreo del espacio de entrada en la raíces del polinomio de orden igual al número de mues-

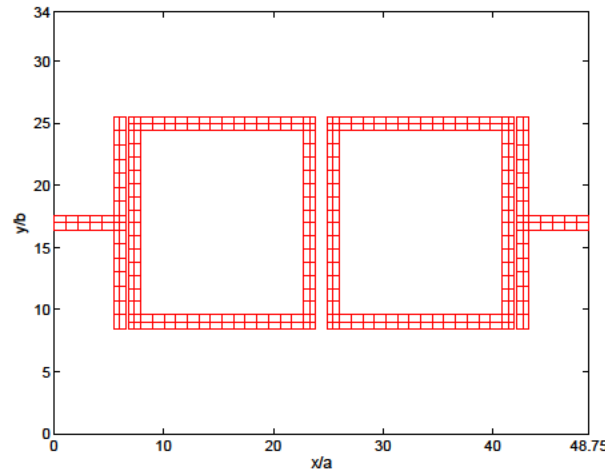


Figura 4.49: Mallado del filtro con anillos cuadrados cerrados.

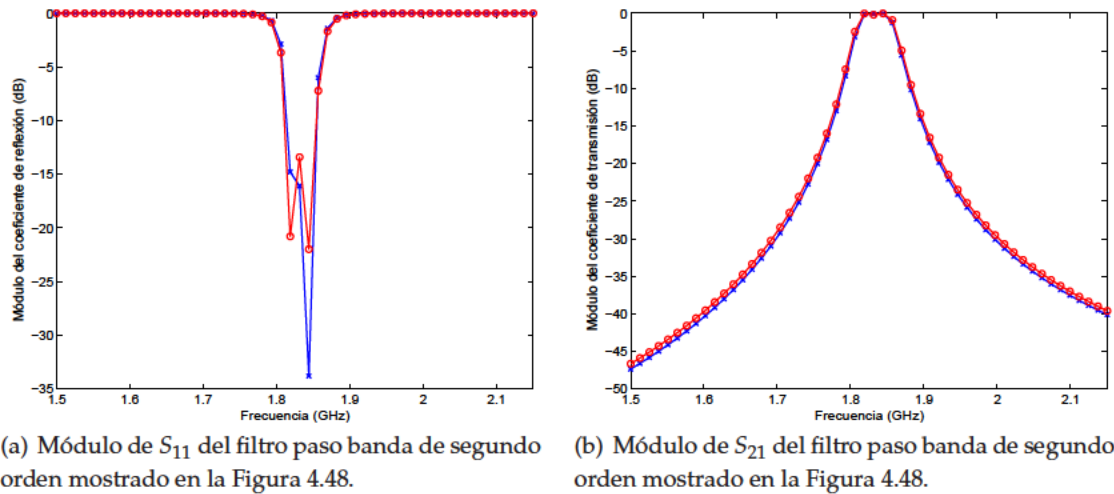


Figura 4.50: Respuesta del filtro de segundo orden de la Figura 4.48 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 52 puntos de frecuencia.

tras. A este polinomio lo hemos denominado polinomio de orden superior. En espacios de entrada multidimensionales se ha optado por funciones neuronales formadas por productos de polinomios. Esta elección permite que la propiedad de ortogonalidad sea cumplida de forma muy efectiva si se muestrea el espacio de entrada en las raíces de cada uno de los polinomios de orden superior de cada una de las coordenadas. El orden dependerá del número de muestras en cada dimensión. La propiedad de ortogonalidad de las funciones neuronales, tanto en espacios unidimensionales como multidimensionales, ha sido probada en casos genéricos.

Se ha demostrado la capacidad de aproximación de las nuevas redes neuronales de forma teórica gracias a la aplicación del teorema de Stone-Weierstrass (ver Apéndice C). Este teorema permite asegurar que un conjunto de funciones constituye un aproximador universal en un espacio de trabajo. En el caso de que el espacio de entrada sea unidimensional, la Red neuronal de Chebychev

equivale a una suma de polinomios de este tipo. Es un hecho conocido que este tipo de funciones son aproximadores universales. Por ejemplo, la fórmula tradicional de aproximación de Chebychev es un aproximador universal que además es una buena aproximación del polinomio minimax. Como se ha demostrado, la Red neuronal de Chebychev unidimensional entrenada con el algoritmo OLS posibilita una aproximación más precisa con el mismo número de términos que la fórmula tradicional. En el caso multidimensional también se ha demostrado que las Redes neuronales de Chebychev son aproximadores universales. Para verificar la capacidad de aproximación se han realizado diferentes pruebas con funciones tanto unidimensionales como multidimensionales. Como muestran los resultados, las Redes neuronales de Chebychev consiguen una precisión similar, e incluso superior para ciertos casos, a las RBFNNs entrenadas con el algoritmo OLS. La principal ventaja de las Redes neuronales de Chebychev estriba en la minimización del número de neuronas.

Para aplicar las nuevas redes neuronales en el problema de aproximación de las funciones de Green es necesario adaptar las zonas fuente y observación a las características de muestreo del espacio de entrada. En efecto, la necesidad de completar la rejilla de puntos de entrenamiento, formada por las raíces de los polinomios de orden superior, impide la utilización de zonas circulares. Todas las zonas fuente y observación deben ser rectangulares. Por ello, las antiguas zonas observación en forma de corona rectangular deben ser divididas en cuatro sub-zonas. Asimismo, las zonas fuente en forma de "L" son divididas en dos zonas rectangulares. Esta obligación conduce a una división mucho más densa del espacio de entrada. No obstante, el tiempo total de selección y entrenamiento de las redes neuronales no crece desmesuradamente gracias a la rapidez del método OLS en este tipo de redes neuronales.

Al igual que en el caso de RBFNNs, la capacidad de aproximación de las funciones de Green se ha comprobado de forma exhaustiva en dos cavidades diferentes. Las Redes neuronales de Chebychev consiguen un descenso progresivo del error conforme aumenta el número de neuronas. En general, una mayor densidad de puntos de entrenamiento conduce a una mejor aproximación. Un hecho sorprendente radica en que para zonas observación reducidas, cerca de la fuente, las nuevas redes consiguen una precisión superior con la técnica de multiplicación por (ρ) . Este hecho se debe a que los productos de los polinomios de mayor orden forman funciones de muy rápida variación, adecuadas para la aproximación de funciones de rápido crecimiento como las funciones de Green multiplicadas por (ρ) . Además, los productos de polinomios son funciones con valores significativos en todo el espacio de trabajo. Esta característica impide la aparición de grandes errores de sobre-entrenamiento. Por el contrario, las funciones gaussianas de las RBFNNs eran similares a deltas de Dirac en la aproximación de funciones de rápido crecimiento, debido al uso de valores de varianza reducidos. De este modo, la aparición del sobre-entrenamiento era más común.

Una vez definidas las nuevas clases de zonas observación y fuente y estudiada la capacidad de aproximación, las Redes neuronales de Chebychev fueron utilizadas con éxito en la aproximación de las funciones de Green en diferentes cavidades. El análisis de diferentes filtros paso banda ha mostrado que las nuevas redes entrenadas con el algoritmo OLS logran una precisión similar a las RBFNNs. El número de neuronas es reducido hasta en un orden de magnitud. Por ello, el análisis circuital es acelerado de modo muy apreciable. El objetivo marcado al comienzo del Capítulo es alcanzado de forma exitosa. La aceleración es tal que el método neuronal de Chebychev es competitivo frente a formulaciones basadas en el dominio espectral. La mejora en el rendimiento frente a este tipo de téc-

nicas es especialmente apreciable en cavidades con un tamaño mayor que las dimensiones del filtro impreso. En este caso, el método neuronal es mucho más rápido y eficaz. El método neuronal de Chebychev es incluso competitivo frente a herramientas comerciales como ADS[®]. De este modo, el tiempo ahorrado en unos pocas decenas de análisis compensa el periodo invertido en la generación de las Redes neuronales de Chebychev.

Otro de los logros de este Capítulo consiste en la ampliación del número de coordenadas del espacio de entrada. Gracias a la ortogonalidad inicial de los regresores, producidos por las neuronas, no es necesario almacenar ninguna matriz en el entrenamiento. El tamaño del conjunto de entrenamiento no se encuentra limitado por la memoria RAM del ordenador. Se pueden generar conjuntos de entrenamiento de decenas de miles de muestras que expresen las variaciones de las funciones de Green respecto a seis o siete variables. En este Capítulo se han mostrado ejemplos de aplicación del método neuronal en cavidades con grosor y permitividad del dieléctrico variables. También se han expuesto diferentes análisis circuitales en una cavidad de dimensiones variables. Sin embargo, el tiempo de selección y sobre todo, el tiempo de entrenamiento de todas las redes neuronales, es muy elevado. Por ello, a pesar de la rapidez del algoritmo de entrenamiento, el método neuronal en cavidades variables es práctico únicamente si se analizan multitud de circuitos. Esta circunstancia puede ocurrir si se diseñan diferentes circuitos. Ciertamente, al aproximar también las variaciones de la cavidad, el método neuronal puede ser utilizado en distintas cavidades ampliando el número de análisis. La metodología aplicada en la aproximación de las funciones de Green realiza una división simple de la coordenada frecuencial. Esta división puede ser insuficiente si aparecen frecuencias de resonancia al cambiar las características de la cavidad. Por tanto, este es un punto a mejorar en futuros trabajos.

El rendimiento óptimo del algoritmo de entrenamiento implica que la mejora del procedimiento de diseño de las redes neuronales debe ejercerse sobre el proceso de generación de los conjuntos de entrenamiento y validación y sobre el proceso de obtención de los regresores. Estos dos procesos consumen la mayor parte del tiempo de diseño de las Redes neuronales de Chebychev. Por tanto, la aplicación de mejores técnicas numéricas en la obtención de las funciones de Green redundará en una mayor eficiencia del método, a pesar de que la solución directa sea a su vez acelerada. La aceleración del cómputo de los regresores es más difícil ya que se han utilizado eficaces fórmulas trigonométricas en el cálculo de los polinomios de Chebychev.

Capítulo 5

Nuevos métodos de diseño de filtros apantallados multicapa

5.1. Introducción

Las herramientas de análisis de circuitos apantallados multicapa desarrolladas en los Capítulos 3 y 4 precedentes permiten una drástica reducción del tiempo de simulación de la respuesta electromagnética. La utilidad de estas nuevas técnicas basadas en redes neuronales va más allá de su aplicación como meras herramientas de análisis de circuitos ya diseñados o fabricados. El uso de los métodos neuronales se puede extender al campo del diseño de circuitos apantallados multicapa. La síntesis de la respuesta ideal de un filtro es un problema que ha sido ampliamente estudiado. En primer lugar se parte de un conjunto de especificaciones. A continuación se debe seleccionar una función de transferencia que satisfaga dichas especificaciones. La función de transferencia elegida suele determinar el orden del filtro. El siguiente paso consiste en hallar los elementos de la matriz de acoplos. A partir de la matriz de acoplos se deben calcular los elementos concentrados que sintetizan la respuesta requerida. En todas las tareas anteriores existen numerosos métodos que posibilitan la obtención de los resultados deseados. La labor final consiste en la plasmación de los elementos concentrados en una tecnología determinada. Por ejemplo en el caso de la tecnología microtira, encapsulada o no, se recurre al uso de resonadores en líneas impresas. Aunque existe una relación, fundamentada en el comportamiento electromagnético de los resonadores, entre los valores de los elementos concentrados y las dimensiones de los resonadores, este problema no es analítico ni fácilmente resoluble en muchas ocasiones. En tecnología microtira no apantallada existen fórmulas empíricas que posibilitan el cálculo de la geometría apropiada. También se pueden generar curvas de diseño que relacionan ciertos parámetros eléctricos con las dimensiones de los resonadores. En cualquier caso, siempre se obtiene un filtro inicial que muy posiblemente no cumpla todas las especificaciones pre-fijadas desde el comienzo del proceso. Es necesario recurrir a la simulación electromagnética del filtro o de algunas de sus partes para ajustar la respuesta actual a la respuesta deseada. Esta fase constituye por tanto un problema de optimización. La reducción del tiempo de análisis aporta diversos beneficios en la labor de la computación de las dimensiones físicas del filtro. En la creación de las curvas de nivel, que relacionan parámetros eléctricos con parámetros geométricos, permitirá la limitación del tiem-

po invertido en esta tarea inicial. Asimismo, si es necesaria una fase de optimización posterior los métodos desarrollados reducirán el tiempo de ajuste.

En este Capítulo se desarrollan dos técnicas diferentes de diseño de filtros impresos en una estructura multicapa apantallada. La primera técnica se basa en la optimización de un filtro inicial con el objetivo de que cumpla una serie de especificaciones pre-fijadas. Para ello se utiliza el método de optimización de los Algoritmos genéticos. Dichos algoritmos son una potente herramienta de optimización basada en las reglas de la evolución de las especies. Esta técnica ha sido empleada de forma limitada en el campo del diseño de filtros encapsulados. Esto es debido a que es un procedimiento que requiere de una cantidad elevada de simulaciones de la respuesta del filtro. A cambio de la alta carga computacional que supone su uso, suelen alcanzar extremos de mejor calidad que las técnicas fundamentadas en el gradiente de la función objetivo. De esta forma, aunque el filtro inicial incumpla claramente las especificaciones, el Algoritmo genético será capaz de alterar sus dimensiones para que al final del proceso se satisfagan las especificaciones. La ausencia de herramientas de simulación suficientemente rápidas ha impedido la extensión de esta clase de técnicas de optimización. Como hemos visto, en los entornos apantallados las técnicas de análisis pueden ser muy lentas. Gracias a los métodos neuronales desarrollados, los Algoritmos genéticos pueden ser aplicados de forma práctica como veremos en la Sección 5.2. Además, se han desarrollado ciertas mejoras en el Algoritmo genético, especialmente orientadas al diseño de filtros, que posibilitan un funcionamiento apropiado y una convergencia correcta hacia la solución.

El segundo método de diseño sigue una vía diferente a la primera técnica. En lugar de optimizar un filtro inicial mediante un procedimiento potente de optimización, busca la generación directamente de un filtro apropiado. La obtención de las dimensiones del filtro se consigue por medio de redes neuronales. En concreto, se han empleado RBFNNs que son entrenadas para aprender las relaciones entre ciertos parámetros eléctricos con las dimensiones de los resonadores que los sintetizan. La tarea que realiza la RBFNN es por tanto la aproximación de las curvas simples y de nivel que sirven a los diseñadores para hallar las dimensiones del filtro. El uso de redes neuronales en esta tarea aporta numerosas ventajas. Las redes neuronales son capaces de interpolar en conjuntos de datos multidimensionales, una labor difícil si se realiza a partir de curvas de nivel. Una vez entrenadas, las RBFNNs devuelven las dimensiones de forma automática e instantánea. De esta forma, liberan al diseñador de esta tarea.

Los parámetros eléctricos que se emplean en el entrenamiento de las RBFNNs provienen de la forma en la que se construye el filtro. Generalmente, a partir de los elementos del prototipo paso bajo, adecuados para sintetizar la función de transferencia elegida, se pueden calcular los factores de calidad externo y coeficientes de acoplo de los resonadores del filtro [Hong y Lancaster, 2001, Cohn, 1957]. Estos parámetros eléctricos junto con la frecuencia de resonancia serán las entradas de la RBFNN. Las salidas serán las dimensiones físicas de los resonadores. En esta técnica, antes de diseñar ningún filtro, es necesario invertir un tiempo en la generación de los conjuntos de entrenamiento y validación adecuados y en el propio entrenamiento de la RBFNN. Como se verá en la Sección 5.3 el tiempo de entrenamiento es de tan solo de unos segundos mientras que el tiempo de generación de los datos de entrenamiento y validación es muy superior. Si se utiliza el método neuronal desarrollado en [Garcia et al., 2006b] el tiempo de generación tarda hasta decenas de horas. Sin embargo, este periodo es reducido hasta unas pocas horas si se emplea el veloz método neuronal basado en las

Redes neuronales de Chebychev. Como se comprobará en la Sección 5.3 el filtro construido con las dimensiones ofrecidas por las RBFNNs no suele satisfacer plenamente las especificaciones. Por ello, es necesario aplicar un proceso de optimización al filtro logrado. No obstante, este filtro constituye un punto de partida de tal calidad que se puede emplear un proceso de optimización por gradiente con resultados óptimos. Más aún, se puede aplicar un proceso de optimización directa gracias tanto a la calidad del filtro inicial como a la rapidez del método de análisis neuronal empleado.

5.2. Método de diseño de filtros apantallados multicapa basado en Algoritmos genéticos

Los Algoritmos genéticos (GA, según las siglas inglesas, AG según las españolas) buscan el extremo global en una determinada región del espacio de búsqueda generado por las variables independientes del problema. Para explorar la región del espacio, el AG opera sobre una población de puntos elegidos inicialmente de modo aleatorio. Cada punto del espacio se llama individuo de la población y está compuesto por las variables independientes del problema. Cada variable tomará un valor determinado en cada individuo, de modo que el conjunto de variables dará lugar a una solución del problema diferente en cada individuo. En todo problema, las variables se deben codificar en los individuos de un modo apropiado para garantizar un correcto funcionamiento del AG. La codificación llevada a cabo y el método aplicado para que la población evolucione correctamente se explica en la siguiente Sub-sección.

5.2.1. Modelo del Algoritmo genético

El AG imita el modo en el que las especies animales consiguen mejorar su capacidad de supervivencia. Este proceso de mejora se produce mediante el intercambio de genes entre los individuos mejor adaptados, es decir, entre aquellos que tienen mayor éxito. De este modo, las variables independientes del problema se codifican en el individuo en forma de genes. En el problema de optimización de un filtro de microondas apantallado, las variables son las dimensiones espaciales (separaciones, longitudes, distancias) que caracterizan el filtro. En nuestro caso, se ha optado por representar en cada gen, bien un incremento o bien un decremento que se debe aplicar sobre un valor inicial de la dimensión en cuestión. El conjunto de todos los genes constituye el cromosoma del individuo, todos los individuos poseen el mismo número de genes y por tanto tienen un cromosoma de igual longitud. Un decremento se puede considerar como un incremento negativo. En el inicio del algoritmo AG se deben definir el valor máximo y mínimo de incremento en cada gen. Generalmente, el valor máximo (x_{max}) del incremento será positivo mientras que el valor mínimo (x_{min}) será negativo. Los valores anteriores permiten limitar las dimensiones del circuito a optimizar, de este modo, se pueden evitar situaciones indeseables como por ejemplo soluciones de fabricación imposible o circuitos que no quepan en la cavidad de apantallamiento. La limitación de las dimensiones circuital es natural en el AG. Para representar los incrementos se ha elegido una codificación de tipo binario. Esta codificación ha sido utilizada a menudo en diversas aplicaciones de los Algoritmos genéticos y posibilita un intercambio de información genética entre individuos sencilla. Para calcular el valor de incremento que se debe aplicar sobre la dimensión inicial se emplea la siguiente ecuación:

$$x = x_{min} + \frac{Ent(gen) \cdot (x_{max} - x_{min})}{2^l - 1} \quad (5.1)$$

donde $Ent(gen)$ es el valor entero sin signo de la cadena de bits que forman el gen y l es el número de bits. En la Figura 5.1 se observa un cromosoma genérico compuesto por dos clases de genes diferentes. Por un lado, existen n genes que codifican variaciones positivas o negativas sobre separaciones entre tiras metálicas de un filtro formado por líneas acopladas (S_1, S_2, \dots, S_n). Por otro lado, existen m genes que codifican variaciones positivas o negativas sobre las longitudes de las tiras metálicas (L_1, L_2, \dots, L_m). Cada gen puede tener un número propio de bits; en este ejemplo todos los genes tienen el mismo número de bits.

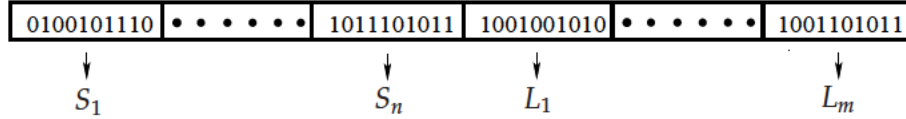


Figura 5.1: Estructura del cromosoma en el problema de diseño de filtros encapsulados. El cromosoma está compuesto por $n + m$ genes. Los n primeros genes codifican las separaciones. Los m últimos genes codifican las longitudes de los resonadores.

El primer paso en el proceso de optimización mediante el AG radica en la creación de la población inicial de individuos. Todos los genes de los individuos son generados de forma aleatoria. La cantidad de individuos es un valor que se fija en el inicio del proceso y que permanece constante a lo largo de todas las iteraciones del algoritmo. Una vez finalizada la inicialización, se sucederán diversas generaciones de nuevos individuos para lograr la evolución de la población. Esta evolución es controlada principalmente por tres mecanismos llamados selección, cruce y mutación. En cada generación se deben aplicar estos mecanismos en orden para producir la nueva población. La selección consiste en la elección de una pareja de individuos (padres) con el objetivo de generar una nueva pareja de individuos (descendencia). Cada individuo posee un valor de rendimiento que indica lo cerca que se encuentra de la solución óptima al problema. Cuanto mayor es el rendimiento de un individuo mayor es la probabilidad de que sea seleccionado como padre. De esta forma, en la nueva población existirá una mayor cantidad de descendientes de aquellos individuos con mayor rendimiento. La selección siempre incorpora cierto grado aleatoriedad de modo que no se pierde toda la información genética de los individuos con menor rendimiento. Éstos pueden también ser seleccionados, aunque con menor probabilidad. La selección de individuos con menor rendimiento posibilita la exploración de extensas zonas del espacio de solución. En nuestro problema se ha escogido la técnica de la ruleta como método de selección [Goldberg, 1989]. En este procedimiento, cada vez que se selecciona un individuo se genera un número aleatorio según una función densidad de probabilidad uniforme. Este número aleatorio posee un valor mínimo igual a la unidad y un valor máximo igual a la suma de los valores de rendimiento de todos los individuos. En la Figura 5.2 se observa la selección de un individuo en una población compuesta por 12 individuos. En el ejemplo el número aleatorio posee un valor de 20, de forma que cae en la zona correspondiente al individuo número 5. Este individuo tiene un valor elevado de rendimiento de modo que será seleccionado con mayor probabilidad ya que posee más números en la ruleta. Para cualquier valor del número aleatorio entre 14 y 21 se elegirá este individuo como uno de los padres de una nueva pareja de descendientes. La zona de valores entre 14 y 21 constituye por tanto la zona de selección del individuo quinto. Si

el rendimiento toma valores reales, se define de igual forma un valor de rendimiento acumulado. Así, para cada individuo se pueden definir los límites inferior y superior que determinan su zona de selección. Existen otros mecanismos de selección como el de torneo que permiten una convergencia similar al algoritmo de la ruleta [Goldberg, 1989]. En las pruebas realizadas no se ha notado una gran diferencia entre estos dos algoritmos de selección. Como complemento del mecanismo de selección se ha implantado la técnica de elitismo. Este mecanismo consiste en la preservación del mejor individuo de una generación. Todos los demás individuos son suplantados por los descendientes, sin embargo, gracias al elitismo el mejor individuo es conservado.

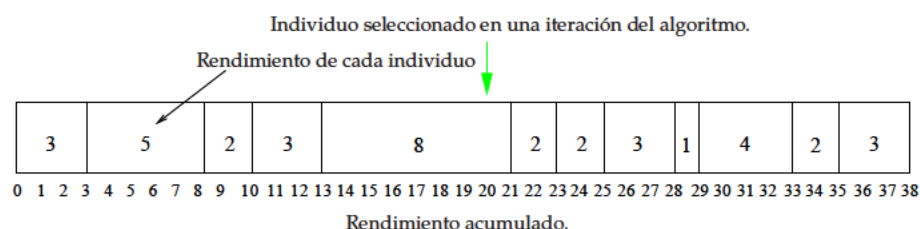


Figura 5.2: Ejecución del algoritmo de la ruleta en la selección de un individuo.

La transmisión del material genético desde los padres a la descendencia se realiza por el mecanismo de cruce. Esta operación se repite en todos los genes de los padres con cierto grado de probabilidad. Se ha elegido un cruce en un solo punto de cada gen. La probabilidad de aplicar el mecanismo de cruce en un gen es relativamente alta, entre un 50 % y un 70 % generalmente. El punto de cruce se elige de forma aleatoria. Si se realiza el cruce en un gen, entonces este punto divide al gen en dos partes. Una parte de los bits del gen de uno de los padres pasa a uno de los hijos mientras que la otra parte de bits pasa al otro hijo. Los bits del gen en los dos hijos se completan con los bits del gen del otro padre. De esta forma, el gen de un hijo estará formado por la parte inicial del gen del primer padre y por la parte final del gen del segundo padre. El otro hijo poseerá un gen con la parte final del gen del primer padre y la parte inicial del segundo padre. El proceso total se aprecia en la Figura 5.3. Si no existe cruce en uno de los genes, entonces cada hijo hereda el gen completo del correspondiente padre como se aprecia en el último gen de la Figura 5.3.

La mutación permite la exploración de nuevas zonas del espacio de solución. Esta técnica produce el cambio del valor de un bit. Al igual que en el mecanismo de cruce, la decisión de mutar un bit es aleatoria. No obstante, la probabilidad de mutación es mucho menor que la probabilidad de cruce. La posibilidad de mutación se aplica en todos los bits que componen el cromosoma y para todos los individuos excepto para aquellos conservados por el elitismo.

5.2.2. Nueva función de rendimiento

La función de rendimiento es uno de los puntos clave en la construcción del AG. En el problema de diseño de un filtro de microondas la función de rendimiento debe medir el grado de cumplimiento de las especificaciones prefijadas. Una función de rendimiento mal definida conducirá a soluciones incorrectas en el AG, es decir, las soluciones encontradas no satisfarán realmente las especificaciones. Uno de los problemas más graves que pueden aparecer durante la ejecución del AG es la convergencia prematura. Este problema se produce cuando algunos individuos de la población obtienen, du-

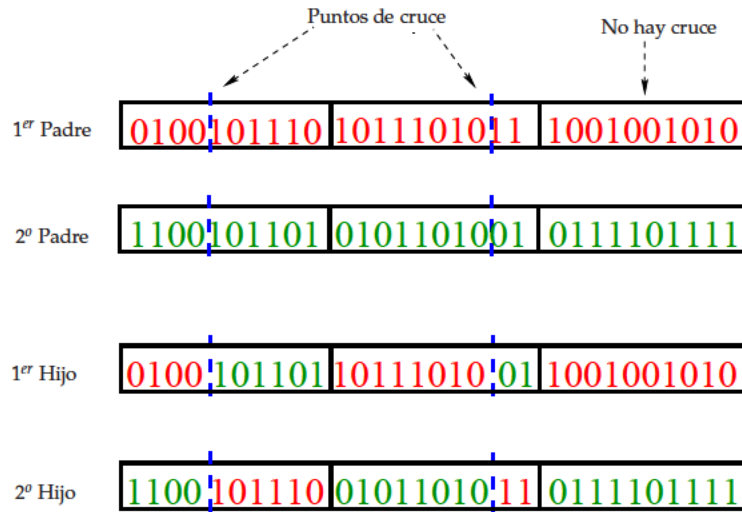


Figura 5.3: Puntos de cruce en un cromosoma compuesto por tres genes.

rante las primeras generaciones, un valor de rendimiento mucho mayor que el resto de la población. Cuando esto ocurre el resto de la población se concentra en torno a estos pocos individuos, de modo que se pierde una de las mayores ventajas del AG, esto es, la capacidad para explorar amplias zonas del espacio de solución. La convergencia prematura provoca que el AG no evolucione hacia nuevas soluciones, de forma que no se produce una optimización efectiva y se pierde la diversidad en las soluciones.

Para definir correctamente el grado de cumplimiento de las especificaciones del circuito y evitar el problema de la convergencia prematura se ha desarrollado una nueva función de rendimiento adecuada. Cada especificación de un filtro consiste generalmente en un nivel de umbral definido para un parámetro S en un determinado ancho de banda o tipo de banda. El conjunto de todos los umbrales constituye la llamada máscara de atenuación. Dependiendo del tipo de parámetro S (reflexión o transmisión) y del tipo de banda (de paso o de rechazo), el valor del parámetro S deberá ser menor o mayor que el umbral. La clave del análisis del rendimiento del filtro radica en la evaluación de la calidad del cumplimiento o incumplimiento de la máscara en cada frecuencia. Por ejemplo, si el parámetro S_{11} debe caer por debajo de -20 dB en una frecuencia en la banda de paso y el valor del parámetro del filtro en esa frecuencia es de -45 dB obviamente cumple la condición. Sin embargo, si el parámetro en otro filtro vale -21 dB también cumple la condición. Efectivamente, no se debe asignar el mismo valor de rendimiento a ambos filtros en esa frecuencia. No obstante, si se gratifica al primer filtro con un valor demasiado elevado de rendimiento, el acierto en esa frecuencia puede enmascarar serios errores en otras frecuencias. Además, la asignación de valores muy elevados, incluso en unas pocas frecuencias, puede conducir a un problema de convergencia prematura. Las frecuencias que cumplan muy claramente las especificaciones del filtro se darán por ejemplo en los polos y ceros, donde el parámetro de reflexión y el de transmisión respectivamente caen bruscamente. La gratificación de la función de rendimiento deberá por tanto saturarse a partir de un cierto límite. Un método para construir una función que cumpla estos requisitos consiste en la utilización de funciones de tipo logístico, ver [Garcia et al., 2006a], como las mostradas en las siguientes ecuaciones:

$$Rend_{menor} = 0,50 \cdot (1 - \tanh(z \cdot (S - U_b))) \quad (5.2a)$$

$$Rend_{mayor} = 0,50 \cdot (1 + \tanh(z \cdot (S - U_b))) \quad (5.2b)$$

La ecuación (5.2a) se debe aplicar si el parámetro S en cuestión debe ser menor que el umbral (U_b), como por ejemplo el parámetro (S_{11}) en la banda de paso. Si el parámetro S debe ser mayor que el umbral, como en el parámetro (S_{21}) en la banda de paso, entonces se debe emplear la ecuación (5.2b). Como se observa en la Figura 5.4 estas funciones crecen a partir del umbral hasta llegar a la saturación. El parámetro z de ambas ecuaciones controla la pendiente de la función logística. El rendimiento total de un filtro es la suma de los rendimientos particulares de los parámetros S en todas las frecuencias de las bandas de interés. Ciertas especificaciones pueden poseer una importancia mayor que el resto. Este grado de importancia se incorpora en el valor total de rendimiento mediante un peso que multiplica a la suma parcial de rendimientos obtenidos para un parámetro S en una determinada banda.

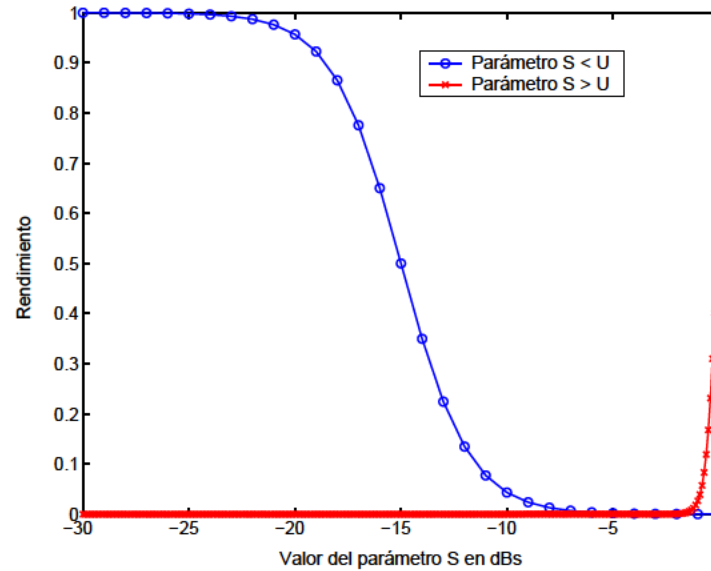


Figura 5.4: Ejemplo de dos funciones de rendimiento con diferentes valores de z (control de la pendiente). La función rendimiento correspondiente a la ecuación (5.2a) se dibuja con círculos. La función rendimiento correspondiente a la ecuación (5.2b) se dibuja con cruces.

Al determinar la pendiente, el parámetro z permite el control de la calidad relativa del cumplimiento de la especificación. Si el parámetro z aumenta entonces la función de rendimiento crece o decrece rápidamente de acuerdo con el cumplimiento o incumplimiento de la especificación respectivamente. Este es el caso correspondiente a la función con cruces de la Figura 5.4. De esta forma, un incumplimiento es penalizado severamente mientras que el cumplimiento se gratifica en gran medida. Por el contrario, si el parámetro z disminuye, la función crece o cae lentamente (función con círculos de la Figura 5.4). En general, valores menores del parámetro z evitan las situaciones de convergencia prematura.

Algunas aplicaciones de los AG usan un método denominado escalado del rendimiento ("Fitness Scaling") para solucionar los problemas de convergencia prematura [Goldberg, 1989]. Con la nueva función de rendimiento no es necesaria ninguna técnica de escalado, ya que se obtiene una buena convergencia del AG de forma directa como mostrarán los ejemplos de la siguiente Sub-sección.

5.2.3. Método de diseño de filtros

El método de diseño propuesto combina la técnica de optimización de los AG y el método neuronal de análisis de circuitos [Garcia et al., 2006a]. En particular se emplea el método basado en RBFNNs [Garcia et al., 2006b]. Este método es suficientemente rápido para acelerar la operación completa del AG. Los AG consumen mucho tiempo en la búsqueda del extremo global debido a que requieren una gran cantidad de ejecuciones de la función a optimizar. Por tanto, es imprescindible disponer de un método de análisis circuital muy rápido para que el AG sea práctico.

El método neuronal aproxima las funciones de Green en una determinada cavidad y para un ancho de banda de trabajo fijo. Por lo tanto, el AG solo podrá operar sobre filtros en una determinada estructura apantallada para la que se disponga del conjunto completo de redes neuronales. No obstante, esta limitación impuesta por el método de análisis es común a otras herramientas de diseño y análisis. Por ejemplo, en el programa comercial *ADS*® un filtro puede ser optimizado para que cumpla ciertas especificaciones mediante diversos métodos de gradiente. Sin embargo, la optimización opera únicamente sobre las dimensiones del filtro de modo que las características de la cavidad no pueden ser cambiadas durante el proceso de optimización. El ancho de banda también es fijado antes del proceso de optimización ya que la máscara de atenuación depende de las bandas de paso y de rechazo. La limitación del ancho de banda en el método neuronal no supone un inconveniente importante ya que se pueden añadir nuevas zonas frecuenciales que extiendan la aplicación del método a nuevas bandas de frecuencia. Como se ha visto en los capítulos precedentes, el ancho de banda de trabajo en el método neuronal es suficientemente amplio para diseñar una gran variedad de filtros. El ahorro de tiempo que se consigue con las redes neuronales respecto a la formulación espacial utilizada permite la aplicación del AG en una cavidad determinada. Como mostrarán los ejemplos de la sub-sección siguiente, tan solo una ejecución del AG para el diseño de un simple filtro convierte en rentable el método neuronal. A continuación se exponen los diferentes pasos que se siguen en el procedimiento de diseño:

1. Seleccionar una estructura multicapa apantallada y un ancho de banda de trabajo. Esta estructura básica permanece inalterada durante todo el proceso.
2. Fijar las especificaciones del circuito (umbrales U_b).
3. Generar las redes neuronales necesarias para aproximar las funciones de Green en la estructura y para el ancho de banda fijado. Para ello aplicar el método neuronal descrito en el Capítulo 3 o en el Capítulo 4.
4. Fijar las dimensiones iniciales del filtro que se va a optimizar. Si el filtro inicial no cumple las especificaciones fijadas se aplica el AG.

5. Fijar los parámetros de la función de rendimiento (pesos y valores de z de la ecuación (5.2b)). Fijar los parámetros del AG, entre otros: probabilidad de cruce, probabilidad de mutación, tamaño de la población, número de bits por gen.
6. Aplicar el AG para optimizar el filtro inicial hasta que, o bien cumpla todas las especificaciones, o bien se alcance el número máximo de iteraciones.

El AG a su vez sigue una serie de procedimientos de forma ordenada para optimizar el filtro inicial:

1. Fijar el rango máximo de variación de las variables independientes del problema (dimensiones del filtro). Son los valores x_{max} y x_{min} en la ecuación (5.1). Esta limitación impide la obtención de soluciones imposibles de fabricar o que excedan los límites de la cavidad. En el AG estos límites se imponen de forma natural en la codificación del gen de la variable correspondiente.
2. Inicialización de la población. Cada individuo es generado mediante la alteración aleatoria de las dimensiones del filtro inicial. La alteración se produce gracias a los incrementos, positivos o negativos, codificados en los genes.
3. Evaluación del rendimiento de los individuos de la población inicial.
4. Comienzo del bucle principal en el que se optimizan los individuos de la población. En cada iteración se realizan las siguientes operaciones:
 - Generación de la nueva población. Se seleccionan n parejas de padres donde n es igual al tamaño de la población menos dos (fijaremos siempre un tamaño de población par). Cada pareja de padres genera dos hijos mediante el intercambio de material genético. Los hijos sustituyen a los dos padres. Los dos mejores individuos son preservados gracias al elitismo.
 - Evaluación del rendimiento de la nueva población.
 - Si algún individuo cumple todas las especificaciones o si se alcanza el número máximo de iteraciones, el algoritmo finaliza. En caso contrario comienza una nueva iteración.

5.2.4. Ejemplos de diseño de filtros apantallados

Para probar la utilidad de la técnica de diseño que combina el AG con el método neuronal se diseñaron distintos filtros en la cavidad mostrada en las Figuras 5.5 y 5.10. Esta cavidad es la usada previamente tanto en el Capítulo 3 (Figura 3.37) como en el Capítulo 4. Las redes neuronales empleadas han sido aquellas generadas con los algoritmos de la Sección 3.8, es decir, con zonas cuadradas y extracción de la singularidad mediante la técnica de multiplicación por (ρ) , ver también [García et al., 2006b]. El ancho de banda de trabajo se sitúa de nuevo entre 9 GHz y 11 GHz.

En primer lugar se procedió al diseño de dos filtros paso banda de segundo orden. Cada filtro optimizado parte de un filtro inicial diferente. La estructura común a ambos filtros se exhibe en la Figura 5.5 mientras que las dimensiones iniciales de los dos filtros se exponen en la Tabla 5.1.

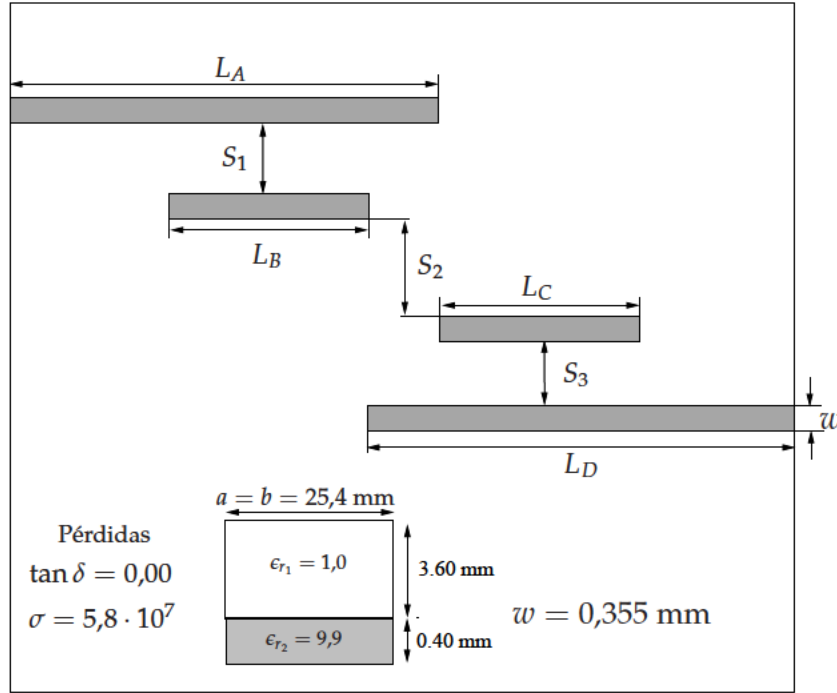


Figura 5.5: Estructura y variables de un filtro de segundo orden formado por líneas acopladas. También se muestran las dimensiones y estructura de la cavidad.

La longitud del puerto de entrada y salida es igual a $L_A = L_D = 14,00 \text{ mm}$ en ambos filtros y permanece inalterada. Se ha aplicado la misma máscara de atenuación en los dos filtros. Así, se busca que el parámetro S_{11} sea menor que -15 dB en la banda de paso $9,95 \text{ GHz} \leq f \leq 10,05 \text{ GHz}$. Por tanto, el ancho de banda relativo es el 1 % y la frecuencia central es 10 GHz . En el parámetro S_{21} se toma un restrictivo valor de umbral de $-0,10 \text{ dB}$ de rizado. Las respuestas de los filtros iniciales se muestran en la Figura 5.8. Se aprecia que ambos se encuentran muy desplazados respecto a la frecuencia central deseada. Además, la atenuación en ambos filtros es mucho mayor que la marcada por la máscara de atenuación. El segundo filtro presenta un rendimiento peor que el primero, por lo que en principio será más difícil de optimizar. Por lo tanto, tal como se observa en la Tabla 5.1, será necesario que el AG opere sobre las longitudes para corregir la frecuencia central y sobre las separaciones entre las tiras metálicas para alcanzar la atenuación deseada (control del acoplo y del ancho de banda).

Tabla 5.1: Dimensiones de los filtros iniciales y optimizados de segundo orden en milímetros.

Variable independiente	Primer filtro inicial	Segundo filtro inicial	Primer filtro optimizado	Segundo filtro optimizado
$L_B = L_C$	6.000	6.050	5.8180	0.0950
$S_1 = S_3$	0.100	0.116	0.1190	0.2247
S_2	0.295	0.263	0.2070	5.8360

La técnica del AG fue aplicada en ambos filtros con el mismo conjunto de parámetros durante 25 generaciones. La población consta de 50 individuos, una probabilidad de cruce del 60 % y una

probabilidad de mutación del 0,030 %. Cada una de las tres variables sobre las que opera el AG fueron codificadas en sendos genes con 10 bits por gen. El AG así desarrollado se ejecutó 5 veces con diferentes poblaciones iniciales para ambos filtros. El parámetro z de la función de rendimiento del parámetro S_{11} fue fijado de forma heurística a un valor de 0,20. Este valor produce una diferencia de rendimiento de 0,10 para una diferencia de 1 dB respecto a la especificación prefijada. Debido a que el margen de valores del parámetro S_{21} que pueden ser gratificados es mucho menor que el margen equivalente del parámetro S_{11} , se eligió un valor más elevado del parámetro z para este parámetro S . En concreto, se optó por un valor de z igual a 3,10 de modo que la pendiente de la función de rendimiento crece de forma más abrupta en el parámetro S_{21} que en el parámetro S_{11} . Así, se produce una caída de 0,15 puntos en el rendimiento cuando existe una diferencia de 0,10 dB entre el valor del parámetro de transmisión del filtro y la atenuación de la especificación. De esta forma, cuando el valor de S_{21} satisface el umbral fijado, la función de rendimiento obtendrá valores significativos. El ajuste de los parámetros z es de vital importancia para asegurar la convergencia correcta de las AG y evitar la aparición de la convergencia prematura. La evolución de las funciones de rendimiento se muestra en la Figura 5.4. Para evaluar la función de rendimiento la banda de paso fue muestreada en 11 puntos equi-distribuidos.

La evolución del rendimiento máximo y medio en el AG para los dos filtros se expone en las Figuras 5.6 y 5.7. Los valores situados sobre las líneas continuas corresponden a la media de las 5 ejecuciones. Asimismo, se exhiben los valores extremos del conjunto de ejecuciones tanto para el valor máximo como para el valor medio de rendimiento. La calidad de la respuesta de los individuos en los dos casos mejora generación a generación. El incremento del rendimiento es más acentuado en el principio del AG. Tras un número relativamente bajo de generaciones se alcanza un máximo global que se sigue optimizando aunque de forma lenta. En el segundo filtro el aumento de la calidad de la respuesta en el inicio del AG es ligeramente más lenta que en el primer filtro debido a que posee una respuesta inicial con peor rendimiento. Sin embargo, al igual que en el primer filtro, en el segundo filtro el máximo global de la optimización se consigue con aproximadamente 10 generaciones. Además, en ambos filtros, a pesar de las diferencias iniciales entre las distintas poblaciones se consigue una evolución satisfactoria en todas ellas. Así, se aprecia en las Figuras 5.6 y 5.7 que a partir de la décima generación los mejores filtros correspondientes a cada una de las cinco ejecuciones tienen un valor de rendimiento prácticamente igual. Este hecho implica que tras diez generaciones del AG todas las poblaciones alcanzan una solución satisfactoria, con un nivel de rendimiento casi igual, aunque la población inicial esté compuesta por filtros poco aptos. Como se observa en la Figura 5.8, la respuesta de los filtros optimizados cumple prácticamente la atenuación fijada en toda la banda de paso. Para apreciar mejor el cumplimiento de la especificación en la Figura 5.9 se muestra la respuesta únicamente en la banda de paso. La atenuación es sobrepasada ligeramente en el comienzo y fin de la banda de paso. También se exhibe la respuesta de los filtros optimizados obtenida con las funciones de Green evaluadas numéricamente (solución directa). Las diferencias entre el método neuronal y la solución directa son pequeñas, de forma que el diseño realizado se puede considerar como válido. La ventaja del método neuronal radica en la aceleración del análisis circuital. El AG emplea una media de 6,73 horas en la ejecución de 25 generaciones completas. Por el contrario, si se utilizase el método exacto, la ejecución de una sola generación tardaría en torno a 51,67 horas. Por tanto, gracias al método neuronal es factible aplicar el AG en la optimización de los filtros de este primer ejemplo.

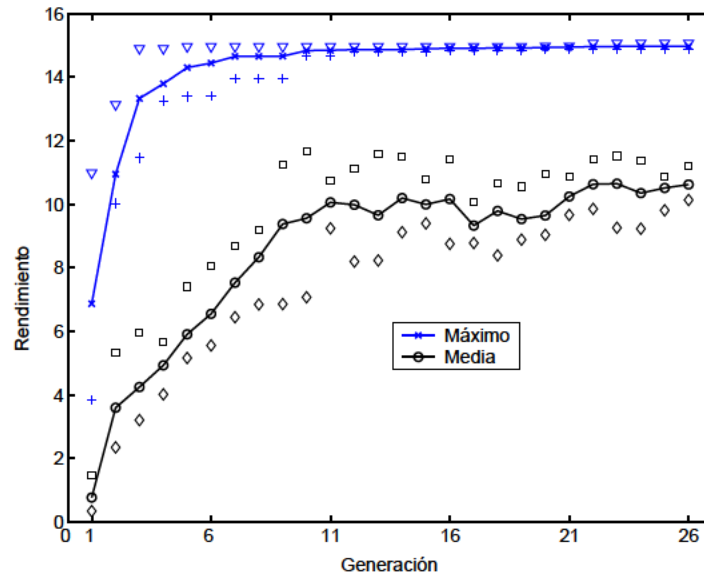


Figura 5.6: Valores máximo y medio del rendimiento correspondiente al primer filtro de segundo orden. Los valores con línea continua son la media de cinco ejecuciones. Los valores sin línea continua son los extremos.

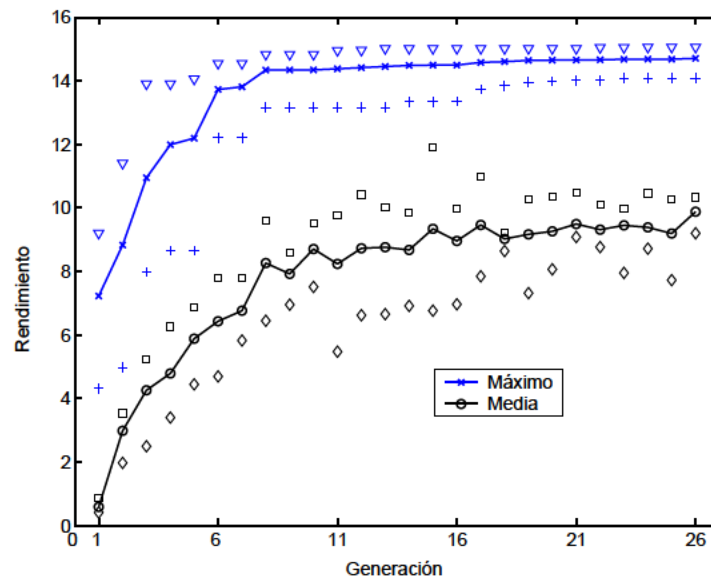


Figura 5.7: Valores máximo y medio del rendimiento correspondiente al segundo filtro de segundo orden. Los valores con línea continua son la media de cinco ejecuciones. Los valores sin línea continua son los extremos.

La técnica propuesta en este Capítulo también fue aplicada al diseño de un filtro paso banda de orden cuarto como el de la Figura 5.10. Las especificaciones son las siguientes: frecuencia central de 10 GHz, un ancho de banda relativo del 6 % y 0,1 dB de rizado en la banda de paso. La primera banda de rechazo se extiende desde los 9 GHz hasta 9,4 GHz mientras que la segunda empieza en 10,6 GHz y finaliza en 11 GHz. En ambas bandas se busca que la atenuación mínima sea de -22 dBs

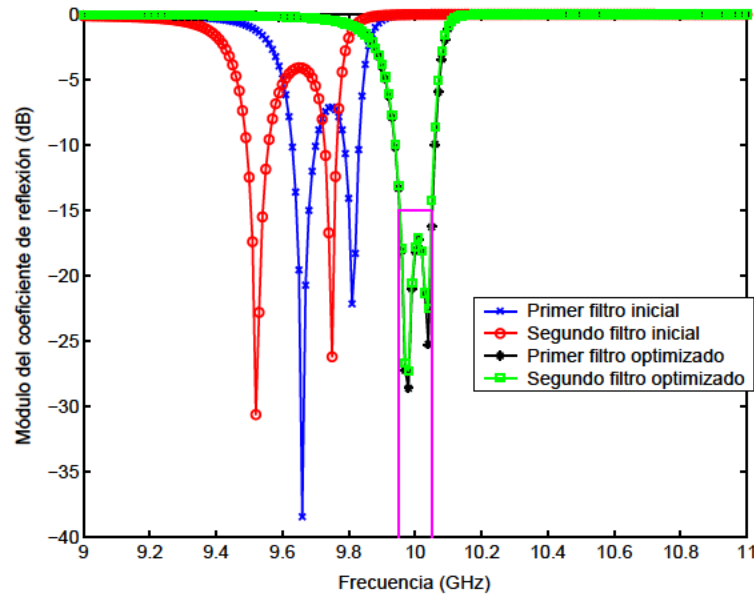


Figura 5.8: Respuesta de los filtros iniciales y optimizados de segundo orden. En línea continua se muestra la máscara de atenuación.

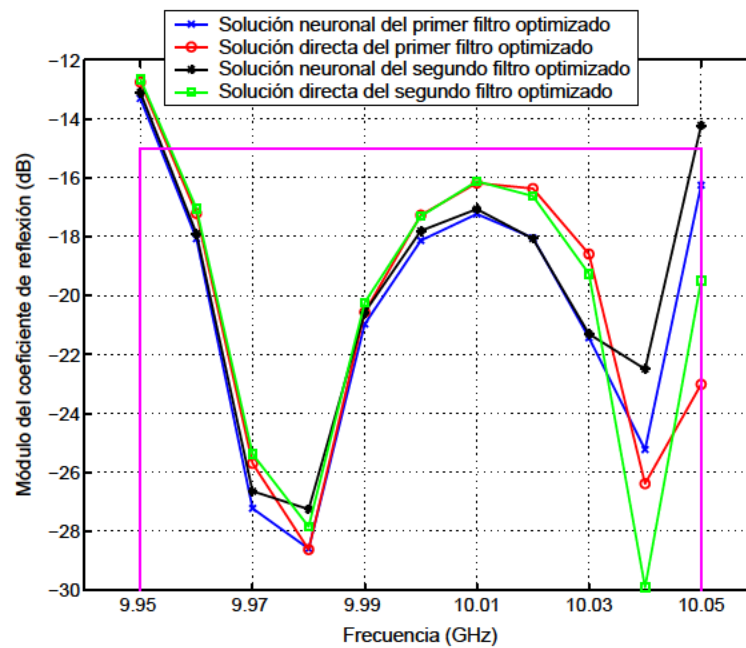


Figura 5.9: Ampliación de la banda de paso de la respuesta de los filtros optimizados de segundo orden. Para validar los resultados se muestran la solución directa obtenida con las funciones de Green evaluadas numéricamente para los dos filtros.

en el parámetro de transmisión. La máscara de atenuación se muestra en la Figura 5.11. En este caso se optó por incluir en el valor de rendimiento total del filtro, el rendimiento en la banda de paso y también el rendimiento producido en las bandas de rechazo. El número total de parámetros z de la función de rendimiento crece hasta un total de seis valores. El enfoque heurístico tomado en los dos filtros de segundo orden ya no es suficiente para obtener una función de rendimiento apropi-

ada. La intervención del diseñador en la fijación de los valores z puede conducir a ineficiencias en el método de diseño. De hecho, una mala elección de los mencionados parámetros puede producir que la excelente convergencia, observada en los ejemplos de segundo orden, se pierda. Incluso en el caso de un diseñador avezado es conveniente liberar al mismo de las tareas de determinación de los parámetros de la técnica de diseño. Por ello, se desarrolló un método para calcular los valores de los distintos parámetros z de forma automática. Este método ajusta los valores de z para que exista una cierta diferencia entre el valor de rendimiento del filtro ideal o deseado con respecto al valor de rendimiento de un filtro auxiliar. Este filtro auxiliar no cumple las especificaciones y puede ser tanto el filtro inicial que se debe optimizar como un filtro cuya respuesta sea apropiada para el propósito del cálculo de los parámetros z . Por ejemplo, en la Figura 5.11 se observa la respuesta ideal de un filtro de cuarto orden de tipo Chebychev y la respuesta de un filtro auxiliar. El primero obtendrá un alto valor de rendimiento mientras que el segundo tendrá un valor menor. De forma visual, un diseñador determina fácilmente en qué medida el filtro auxiliar cumple las especificaciones. El valor de rendimiento expresa de forma numérica ese grado de cumplimiento. El modo de cuantificar dicho grado será de forma relativa al rendimiento del filtro ideal. Además, el ajuste de la diferencia entre los dos rendimientos permite que otros filtros excelentes no tomen un valor de rendimiento desmesurado respecto al resto de la población. De esta forma se evita el problema de la convergencia prematura.

En general, en el diseño de un filtro tendremos que optimizar una serie de n respuestas: $(S_1, S_2, S_3, \dots, S_n)$ en un conjunto de bandas de frecuencia: $(\Delta_1, \Delta_2, \Delta_3, \dots, \Delta_m)$. Los parámetros z que se deben computar son:

$$\mathbf{z} = (z_{(S_1, \Delta_1)}, z_{(S_2, \Delta_1)}, \dots, z_{(S_n, \Delta_1)}, \\ z_{(S_1, \Delta_2)}, z_{(S_2, \Delta_2)}, \dots, z_{(S_n, \Delta_m)}, \dots, \\ z_{(S_1, \Delta_m)}, z_{(S_2, \Delta_m)}, \dots, z_{(S_n, \Delta_m)}) \quad (5.3)$$

El rendimiento de la respuesta i -ésima S_i en la banda j -ésima Δ_j se calcula mediante la ecuación (5.4). El signo del término \pm dependerá del tipo de respuesta. Si la respuesta i -ésima debe ser menor que el valor de umbral se empleará el signo $-$, si por el contrario debe ser mayor, entonces se utilizará el signo $+$. La función de rendimiento puede ser multiplicada por un peso $\beta_{(i,j)}$ que pondere su importancia en la expresión total del rendimiento.

$$Rend(z_{(S_i, \Delta_j)}) = \sum_{f \in \Delta_j} 0,50 \cdot \beta_{(i,j)} \cdot \left(1 \pm \tanh \left(z_{(S_i, \Delta_j)} \cdot (S_i(f) - U_{b(S_i, \Delta_j)}) \right) \right) \quad (5.4)$$

Una vez determinado el signo apropiado de cada función de rendimiento particular se calcula el rendimiento total del filtro ideal y del filtro auxiliar gracias a las ecuaciones (5.5a) y (5.5b) respectivamente:

$$\begin{aligned}
Rend_total_{circ_ideal}(z) = & Rend_{circ_ideal}(z_{(S_1, \Delta_1)}) + Rend_{circ_ideal}(z_{(S_2, \Delta_1)}) + \dots + Rend_{circ_ideal}(z_{(S_n, \Delta_1)}) \\
& + Rend_{circ_ideal}(z_{(S_1, \Delta_2)}) + Rend_{circ_ideal}(z_{(S_2, \Delta_2)}) + \dots + Rend_{circ_ideal}(z_{(S_n, \Delta_2)}) \\
& + Rend_{circ_ideal}(z_{(S_1, \Delta_m)}) + Rend_{circ_ideal}(z_{(S_2, \Delta_m)}) + \dots + Rend_{circ_ideal}(z_{(S_n, \Delta_m)})
\end{aligned} \tag{5.5a}$$

$$\begin{aligned}
Rend_total_{circ_aux}(z) = & Rend_{circ_aux}(z_{(S_1, \Delta_1)}) + Rend_{circ_aux}(z_{(S_2, \Delta_1)}) + \dots + Rend_{circ_aux}(z_{(S_n, \Delta_1)}) \\
& + Rend_{circ_aux}(z_{(S_1, \Delta_2)}) + Rend_{circ_aux}(z_{(S_2, \Delta_2)}) + \dots + Rend_{circ_aux}(z_{(S_n, \Delta_2)}) \\
& + Rend_{circ_aux}(z_{(S_1, \Delta_m)}) + Rend_{circ_aux}(z_{(S_2, \Delta_m)}) + \dots + Rend_{circ_aux}(z_{(S_n, \Delta_m)})
\end{aligned} \tag{5.5b}$$

Los parámetros del vector z se calculan de modo que se cumpla la relación (5.6a). El ratio entre los dos rendimientos es fijado por el diseñador. Las ecuaciones obtenidas son no-lineales; una forma de calcular los parámetros z consiste en tratar el problema como una labor de minimización de la relación equivalente mostrada en la ecuación (5.6b). Este problema de optimización es sencillo y la convergencia hacia una solución satisfactoria es rápida.

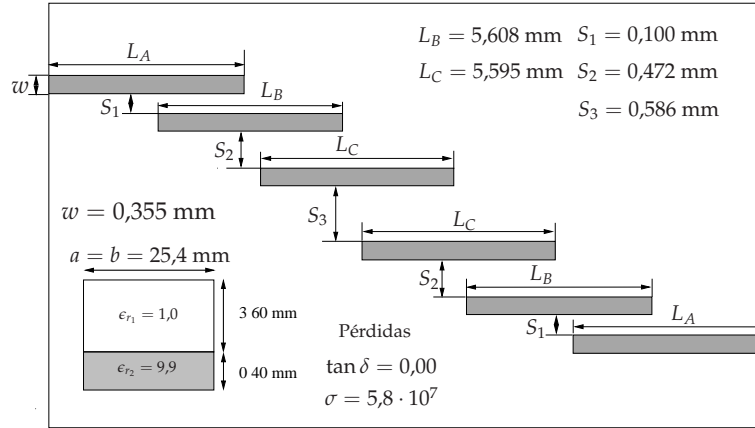
$$Rend_total_{circ_ideal}(z) = Ratio \cdot Rend_total_{circ_aux}(z) \tag{5.6a}$$

$$Rend_total_{circ_ideal}(z) - Ratio \cdot Rend_total_{circ_aux}(z) = 0 \tag{5.6b}$$

Si el filtro es de tipo paso banda y las especificaciones se refieren a las respuestas de los coeficientes de reflexión y transmisión, entonces existen un total de seis parámetros z (ver ecuación (5.7a)). Siguiendo el procedimiento anterior, en primer lugar se obtienen las expresiones del rendimiento del filtro ideal y del filtro auxiliar con las ecuaciones mostradas en (5.7b) y (5.7c) respectivamente. Estas expresiones se aplicaron sobre los filtros y umbrales de la Figura 5.11 con el objetivo de diseñar un filtro de cuarto orden. El ratio entre el valor de rendimiento de la respuesta ideal y de la respuesta del filtro auxiliar fue fijado en un valor igual a 1,50. La minimización de la relación (5.7e) conduce a la obtención de los valores de la Tabla 5.2. Se utilizó la función *fmincon* de *MATLAB*®. Esta función emplea una estrategia de subespacio de región de confianza basado en el método interior-reflexivo de Newton. En todos los parámetros z se impusieron diferentes restricciones sobre los valores finales. De este modo, en la banda de paso los parámetros pueden tener un valor entre 0 y 5 mientras que en las bandas rechazadas el rango de variación se extiende desde 0 hasta 1. Estos parámetros producen las funciones de rendimiento exhibidas en la Figura 5.12. El valor extremo de $z_{(S_{21}, primera_banda_rechazo)} = 0,0013$ produce una función de rendimiento prácticamente plana, sin discriminación entre el cumplimiento y el incumplimiento de la especificación. Sin embargo, el uso de los parámetros z en conjunto produce un valor de rendimiento adecuado. Se han utilizado dos tipos de pesos en las funciones de rendimiento para marcar su importancia en el cálculo del rendimiento total. En la banda de paso, las funciones de rendimiento emplean un peso igual a 2,25. En las banda eliminadas el peso baja hasta un valor de 0,50.

Tabla 5.2: Valor de los parámetros z para los filtros de la Figura 5.10.

Parámetro z	Valor
$Z(S_{11}, \text{banda_paso})$	4.3313
$Z(S_{21}, \text{banda_paso})$	2.1237
$Z(S_{11}, \text{primera_banda_rechazo})$	0.3437
$Z(S_{21}, \text{primera_banda_rechazo})$	0.0013
$Z(S_{11}, \text{segunda_banda_rechazo})$	0.5900
$Z(S_{21}, \text{segunda_banda_rechazo})$	0.5100

**Figura 5.10:** Estructura y variables de un filtro de cuarto orden formado por líneas acopladas. Se muestran las dimensiones de filtro inicial que debe ser optimizado. También se muestran las dimensiones y estructura de la cavidad.

$$\mathbf{z} = (z(S_{11}, \text{banda_paso}), z(S_{21}, \text{banda_paso}), z(S_{11}, \text{primera_banda_rechazo}), z(S_{21}, \text{primera_banda_rechazo}), z(S_{11}, \text{segunda_banda_rechazo}), z(S_{21}, \text{segunda_banda_rechazo})) \quad (5.7a)$$

$$\begin{aligned} Rend_total_{circ_ideal}(z) = & Rend_{circ_ideal}(z(S_{11}, \text{banda_paso})) + Rend_{circ_ideal}(z(S_{21}, \text{banda_paso})) \\ & + Rend_{circ_ideal}(z(S_{11}, \text{primera_banda_rechazo})) + Rend_{circ_ideal}(z(S_{21}, \text{primera_banda_rechazo})) \\ & + Rend_{circ_ideal}(z(S_{11}, \text{segunda_banda_rechazo})) + Rend_{circ_ideal}(z(S_{21}, \text{segunda_banda_rechazo})) \end{aligned} \quad (5.7b)$$

$$\begin{aligned} Rend_total_{circ_aux}(z) = & Rend_{circ_aux}(z(S_{11}, \text{banda_paso})) + Rend_{circ_aux}(z(S_{21}, \text{banda_paso})) \\ & + Rend_{circ_aux}(z(S_{11}, \text{primera_banda_rechazo})) + Rend_{circ_aux}(z(S_{21}, \text{primera_banda_rechazo})) \\ & + Rend_{circ_aux}(z(S_{11}, \text{segunda_banda_rechazo})) + Rend_{circ_aux}(z(S_{21}, \text{segunda_banda_rechazo})) \end{aligned} \quad (5.7c)$$

$$Rend_total_{circ_ideal}(z) = Ratio \cdot Rend_total_{circ_aux}(z) \quad (5.7d)$$

$$Rend_total_{circ_ideal}(z) - Ratio \cdot Rend_total_{circ_aux}(z) = 0 \quad (5.7e)$$

Una vez calculados los parámetros del vector \mathbf{z} hay que fijar los parámetros del AG, determinar las dimensiones del filtro inicial que se debe optimizar y fijar las variables independientes y su rango

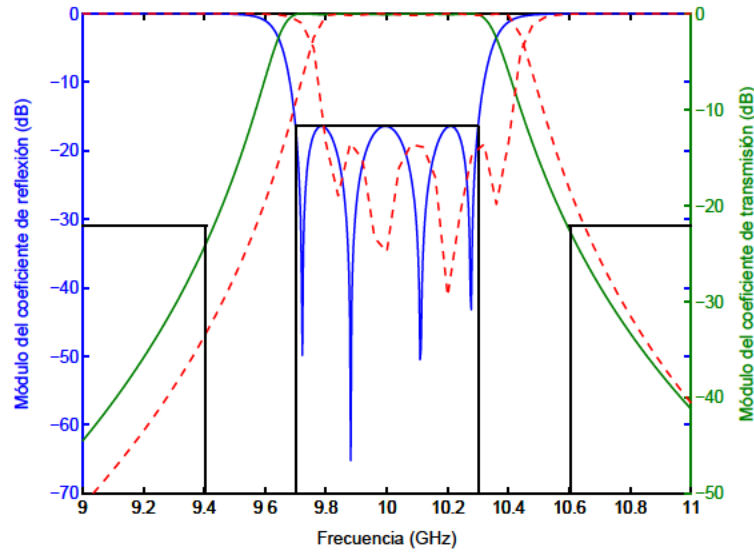
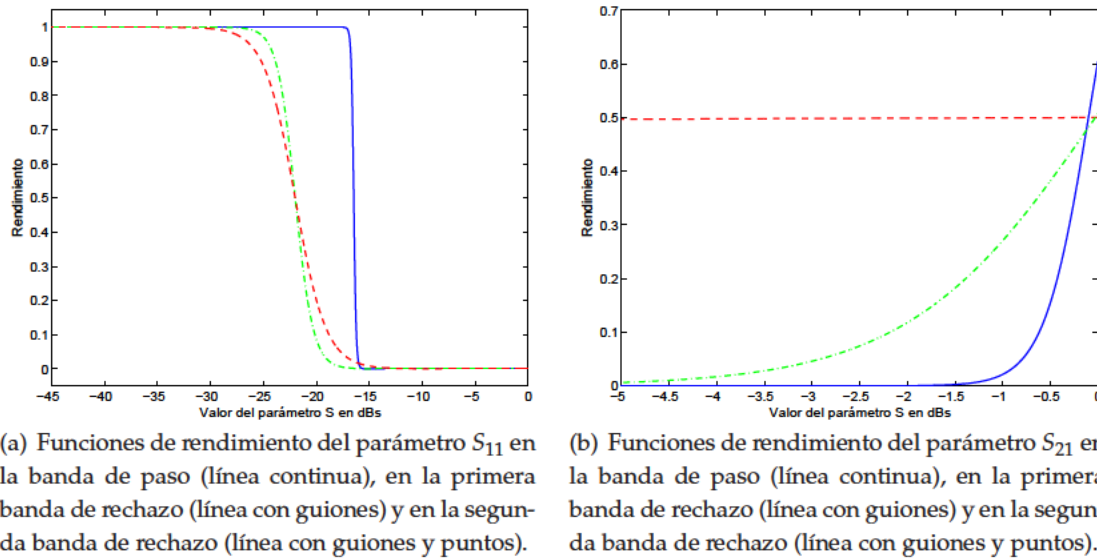


Figura 5.11: Respuesta de un filtro ideal de tipo Chebychev de cuarto orden (línea continua) y respuesta de un filtro auxiliar (línea con guiones) para calcular los parámetros z de las funciones de rendimiento. El rizado en la banda de paso es 0,10 dB. Se muestra la máscara de atenuación con líneas continuas.



(a) Funciones de rendimiento del parámetro S_{11} en la banda de paso (línea continua), en la primera banda de rechazo (línea con guiones) y en la segunda banda de rechazo (línea con guiones y puntos).

(b) Funciones de rendimiento del parámetro S_{21} en la banda de paso (línea continua), en la primera banda de rechazo (línea con guiones) y en la segunda banda de rechazo (línea con guiones y puntos).

Figura 5.12: Funciones de rendimiento de los parámetros S_{11} y S_{21} correspondientes a los valores de parámetros z recogidos en la Tabla 5.2.

de variación. Las dimensiones iniciales del filtro se muestran en la Figura 5.10. La respuesta simulada con el método neuronal y la solución directa para este filtro se muestran en la Figura 5.14. Existen cinco variables sobre las que opera el AG: las separaciones entre resonadores y puertos de entrada y salida (S_1, S_2 y S_3) y las longitudes de los resonadores (L_B y L_C). De nuevo, se fijó una probabilidad de cruce del 60 %, una probabilidad de mutación del 0,03 % y una codificación de 10 bits en cada gen. La población estuvo compuesta por 100 individuos que se optimizan durante 25 generaciones como máximo. El ancho de banda de trabajo es muestreado en 51 puntos para todos los individuos

de la población. Con esta serie de parámetros una ejecución del AG dio como resultado la evolución expuesta en la Figura 5.13. El problema de optimización del filtro de cuarto orden es más complicado que el de optimización de los filtros de segundo orden. La mejora de la media de la población es firme pero la obtención de un mejor filtro en la población de individuos requiere de varias generaciones. No obstante, al final del proceso se alcanza un filtro con un rendimiento muy superior al mejor conseguido en la población inicializada aleatoriamente. Este valor de rendimiento superior refleja efectivamente un cumplimiento más completo de las especificaciones. Como se aprecia en la Figura 5.15 la respuesta del filtro final cumple prácticamente todas las especificaciones; es un filtro de una calidad muy superior a la del filtro inicial. En la Figura 5.15 también se muestra la solución directa obtenida con la IE y las funciones de Green evaluadas numéricamente. Existen pequeñas diferencias entre el método neuronal y dicha solución debido a la existencia de acoplos críticos entre los puertos de entrada y salida y los correspondientes resonadores, como se aprecia en la Tabla 5.3.

En este ejemplo, la función de rendimiento ha probado su capacidad para expresar correctamente el grado de cumplimiento de las especificaciones. Además, evita la aparición de problemas debidos a la convergencia prematura. En la Figura 5.16(a) se expone el rendimiento de los individuos de la población inicializada aleatoriamente, es decir, de la primera generación de la ejecución del AG. Se observa que aparece un individuo con un rendimiento superior al resto. Si este filtro obtuviera un valor de rendimiento demasiado grande sería seleccionado muchas veces como padre en la siguiente generación de forma que casi toda la nueva población estaría compuesta por sus descendientes. Se perdería así toda capacidad de exploración del espacio de solución. La función de rendimiento permite asignar un valor de rendimiento suficiente para indicar que ese individuo tiene propiedades positivas pero sin que ese valor suponga una gran diferencia con el resto de individuos. En efecto, el valor asignado es un poco más de dos veces mayor que la media (19,757) y cerca de cinco veces mayor que el mínimo valor de rendimiento (8,776). La ausencia de convergencia prematura se aprecia en la Figura 5.16(b). De hecho, el individuo con mayor rendimiento es elegido en tan solo en una ocasión como padre. Se podría dar incluso el caso de que no fuera seleccionado en ninguna ocasión como padre. Este hecho se debe a que la población es relativamente grande y el rendimiento del mejor individuo aunque grande solo representa el 2 % del rendimiento total acumulado (1975,7). Esta situación tampoco es deseable ya que implica la pérdida del contenido genético del mejor individuo en la siguiente generación. La aplicación de la técnica del elitismo corrige este funcionamiento erróneo, de forma que los dos mejores individuos son preservados de una generación a la siguiente. Así, gracias al elitismo el contenido genético de los dos mejores individuos de la primera generación no se perdería aunque no fueran seleccionados como padres en ningún emparejamiento. Gracias a la preservación, estarían incluidos en la segunda generación. Por tanto, podrían ser seleccionados como padres para generar los individuos de la tercera generación, de forma que sus genes, es decir, sus características se expandirían en la población aunque no fueran padres de individuos de la segunda generación.

Como se ha visto en esta sub-sección la nueva función de rendimiento permite la correcta expresión del cumplimiento de las especificaciones. Los parámetros de dicha función se calculan mediante un procedimiento automático que carece prácticamente de las debilidades propias de un ajuste heurístico. Asimismo, esta función evita los problemas de convergencia prematura. Los beneficios de la nueva función de rendimiento junto con la apropiada selección de los parámetros de AG han

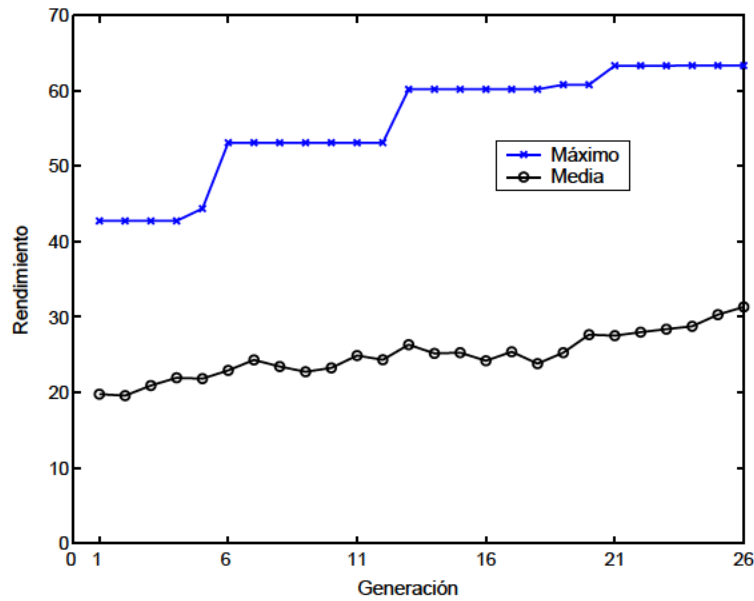


Figura 5.13: Valores máximo y medio del rendimiento correspondiente al filtro de cuarto orden.

Tabla 5.3: Dimensiones del filtro inicial y optimizado de cuarto orden en milímetros. L_A permanece constante y es igual a 7,538 mm.

Variable independiente	Filtro inicial	Filtro optimizado	Incremento
L_B	5.708	5.6036	-0.10400
L_C	5.772	5.5957	-0.17630
S_1	0.130	0.0693	-0.06068
S_2	0.462	0.4685	-0.00586
S_3	0.546	0.5689	0.02293

permitido alcanzar una solución a cada problema de optimización en un número de generaciones relativamente bajo. En los filtros de dos polos la solución se consigue con 10 generaciones mientras que en el problema más complicado de optimización del filtro de cuarto orden se necesitan 20 generaciones. Conforme aumenta la dificultad del problema de diseño es más probable que el mejor individuo de la población de la última generación no cumpla todas las especificaciones. Así ha ocurrido en el diseño del filtro de cuarto orden. En este caso se ha obtenido un filtro de gran calidad pero que incumple las especificaciones en ciertas frecuencias de interés. Una solución consiste en utilizar un rápido método de optimización por gradiente que optimice los mejores filtros conseguidos en la última generación. Estos filtros poseen prestaciones cercanas al filtro ideal, por lo que unas pocas iteraciones de un algoritmo de optimización basado en el uso del gradiente debería producir un filtro plenamente satisfactorio. Un segundo problema en la ejecución del AG radica en la obtención del filtro inicial que se debe optimizar. En los ejemplos hemos partido de filtros con una respuesta degradada gracias a la variación de las dimensiones de filtros que sí cumplían ciertas especificaciones. Por ejemplo, el filtro de cuarto orden inicial se obtuvo a partir del filtro de la Figura 3.38. Se ha procedido de este modo para probar la utilidad y rendimiento del AG. En un caso real de diseño,

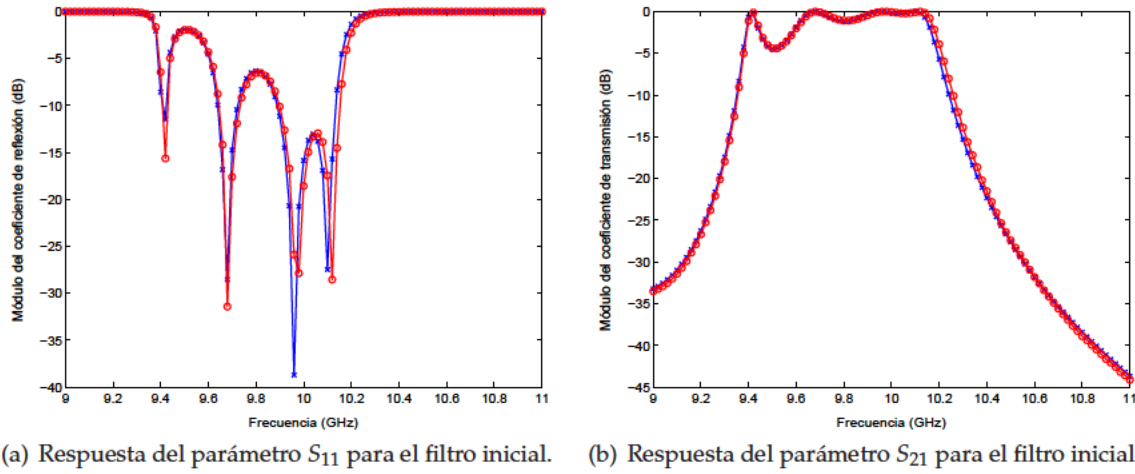


Figura 5.14: La respuesta neuronal se dibuja con cruces, la solución directa obtenida con las funciones de Green evaluadas numéricamente se dibuja con círculos.

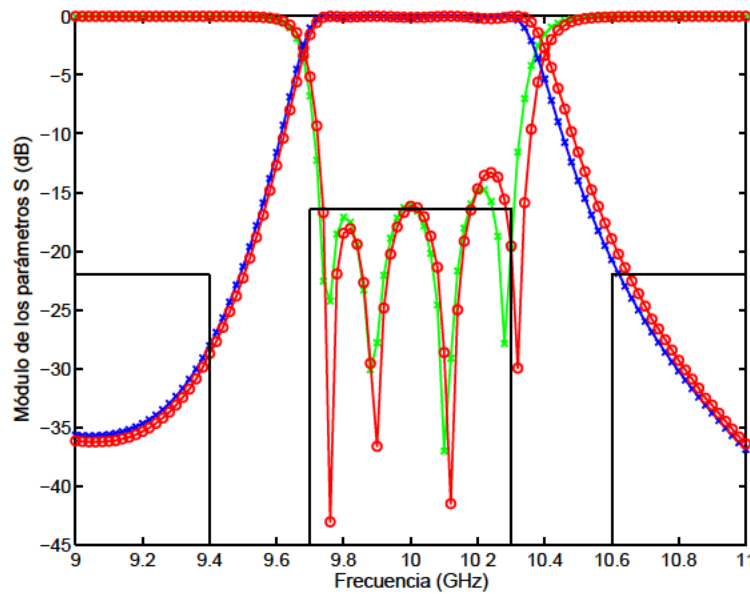


Figura 5.15: Respuesta del filtro optimizado de cuarto orden. La respuesta neuronal se dibuja con cruces, la solución directa obtenida con las funciones de Green evaluadas numéricamente se dibuja con círculos. En línea continua se muestra la máscara de atenuación.

se puede partir de un filtro conocido cuya respuesta no se encuentre alejada de la deseada. En estos casos el AG puede carecer de utilidad debido a que emplea una gran cantidad de análisis circuitales. Las técnicas de gradiente o incluso el ajuste llevado cabo por un diseñador serían suficientes para encontrar las dimensiones del filtro deseado. Por tanto, el AG será útil si el filtro inicial presenta una respuesta alejada de la deseada. La técnica de diseño de la siguiente sección constituye una solución al problema de la búsqueda de un filtro inicial. La siguiente técnica intenta encontrar directamente las dimensiones del filtro que cumple las especificaciones. Si las dimensiones halladas producen una respuesta que no satisface totalmente las especificaciones, entonces el filtro encontrado constituye un punto inicial para un fácil proceso de optimización.

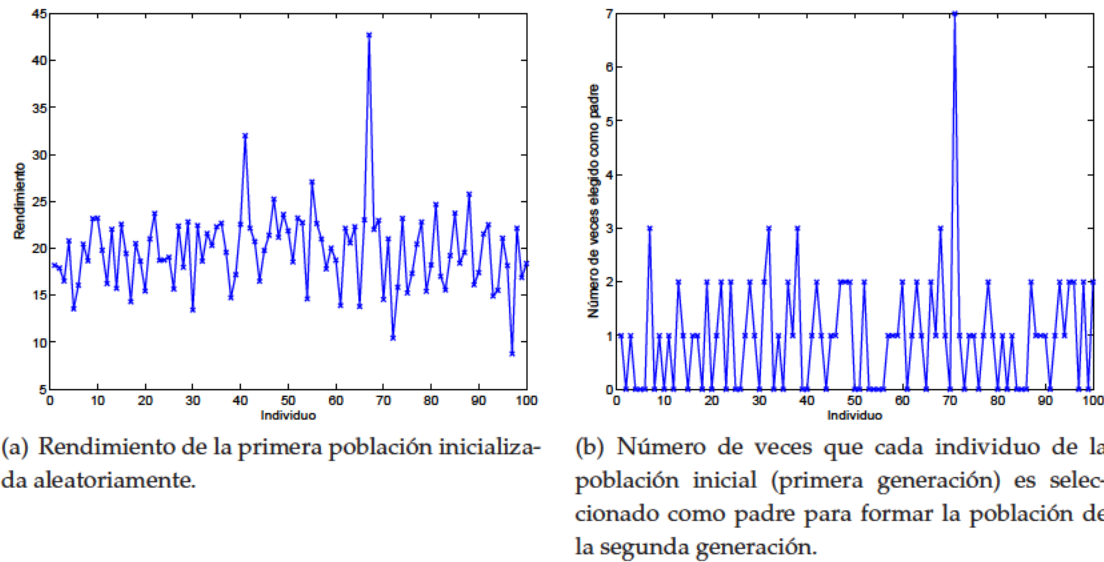


Figura 5.16: Características de la población inicial (primera generación) en el diseño del filtro de cuarto orden.

5.3. Método de diseño de filtros apantallados multicapa basado en redes neuronales

5.3.1. Introducción

Aunque las primeras aplicaciones de las redes neuronales se centraron en el modelado de distintos dispositivos de microondas, existen un buen número de aplicaciones en el diseño de circuitos de microondas. Por ejemplo, en [Selleri et al., 2002] las redes neuronales se usaron para diseñar una antena de tipo bocina y en [Fedi et al., 1999] se generaron nuevos filtros gracias a redes neuronales de clase MLP. Los modelos neuronales que simulan la respuesta electromagnética de un circuito a partir de las dimensiones geométricas se denominan modelos o problemas directos. Por ello, al problema de diseño, que implica la obtención de las dimensiones físicas a partir de la respuesta deseada, se le llama problema inverso. Las redes neuronales se pueden emplear en la labor inversa de dos formas diferentes. En el primer procedimiento, llamado método de optimización, el modelo neuronal directo es evaluado de forma repetida para encontrar las dimensiones físicas que permiten un buen ajuste entre la respuesta buscada y la simulada. El modelo neuronal directo puede emplearse como una caja negra que no se altera durante el proceso de optimización. En este caso, únicamente se actúa sobre las entradas de la primera capa sin cambiar los parámetros de la red neuronal. Otra opción consiste en usar las redes neuronales del modelo directo en un modo inverso. En este método, el proceso de optimización actúa, no solamente sobre las entradas, sino también en el interior del modelo neuronal. Así, en [Vai et al., 1998], el error entre la salida obtenida con unas entradas iniciales y la salida deseada es retro-propagado en un MLP. En lugar de cambiar los pesos de las neuronas, como ocurriría en el entrenamiento directo, el gradiente del error actúa sobre las entradas neuronales. El cambio en las entradas alcanza la capa de salida proporcionando de esta forma las dimensiones deseadas. Si la salida correspondiente a las nuevas entradas no genera la salida deseada se llevan a cabo nuevos ciclos. En el segundo enfoque, diferente del método de optimización, no se usan las

redes neuronales empleadas en el modelado directo del dispositivo sino que se generan una serie de redes neuronales dedicadas al problema de diseño. Así, las entradas de estas nuevas redes neuronales corresponden a la respuesta eléctrica y las salidas son las dimensiones físicas buscadas. De este modo, no es necesario utilizar el modelo neuronal repetidas veces para alcanzar una solución aceptable. A esta metodología se le suele denominar modelado inverso directo. En [Wang et al., 2006] y [Kabir et al., 2008] esta estrategia fue aplicada en la aproximación de las relaciones entre los elementos de una matriz de acoplos y las dimensiones físicas en una estructura de guíaonda que sintetiza los mencionados elementos. La técnica resultante fue empleada en el diseño de diversos filtros pseudo-elípticos. Las redes neuronales también han sido aplicadas en el diseño de filtros impresos en una estructura multicapa. Así, en [Watson et al., 1999] una red MLP fue usada para obtener las dimensiones geométricas de filtros formados por líneas acopladas usando como entradas ciertos parámetros modales de tipo normal, como por ejemplo la impedancia o los ratios de voltaje ("voltage ratios").

La técnica propuesta para diseñar filtros encapsulados multicapa mediante redes neuronales es similar a la empleada en [Wang et al., 2006] y [Kabir et al., 2008]. En nuestro caso, las RBFNNs se encargan de la aproximación de las relaciones entre los factores de calidad externos, los coeficientes de acoplo y frecuencia de resonancia, y las dimensiones de los resonadores que los sintetizan. Las variaciones de los parámetros de reflexión y transmisión quedan determinadas por un tipo de función de transferencia. El uso de funciones como la de Chebychev, la de Butterworth o la elíptica está muy extendido. La función de transferencia permite calcular el orden que debe tener el filtro para que se cumplan las especificaciones. Además, los coeficientes del prototipo paso bajo posibilitan la evaluación de los factores de calidad y coeficientes de acoplo de una manera sencilla [Matthaei et al., 1980]. La principal dificultad radica en el cálculo de las dimensiones de los resonadores que sintetizan los parámetros eléctricos computados. En el caso de filtros de tecnología microtira impresos en substratos, en los que se consideran dimensiones transversales infinitas, existen fórmulas empíricas que permiten evaluar de forma aproximada las dimensiones físicas mediante tablas. Verbigracia, en el caso de resonadores de líneas acopladas las dimensiones son calculadas a partir de los valores de impedancia par e impar [Hong y Lancaster, 2001]. En otros casos, las dimensiones físicas del filtro son obtenidas a partir de curvas de diseño que relacionan dichas dimensiones con los factores de calidad, coeficientes de acoplo y otros parámetros eléctricos, [Hong y Lancaster, 2001, Weller, 2000, Kuo y Cheng, 2004].

En los filtros multicapa apantallados las dimensiones transversales son finitas y no existen fórmulas empíricas que calculen fácilmente las dimensiones circuitales. Si se emplean las ecuaciones empíricas correspondientes a la estructura no apantallada, la precisión requerida no se suele alcanzar. Este hecho es debido a que los efectos e interacciones con las paredes metálicas son despreciados. Una solución consiste en la construcción de nuevas curvas de diseño para este tipo de filtros. Sin embargo, la obtención de estas nuevas curvas requiere de la evaluación de un elevado número de factores de calidad y coeficientes de acoplo para conseguir suficiente precisión en la tarea de diseño. Estos cálculos se deben realizar mediante técnicas de análisis como la IE o el FEM. Tales procedimientos consumen una gran cantidad de recursos informáticos de modo que la construcción de las curvas de diseño se convierte en una tarea difícil y laboriosa.

La técnica desarrollada en este trabajo evita los dos inconvenientes principales del método de

las curvas de diseño, conservando a la vez sus ventajas. Las RBFNNs aprenden las relaciones entre los parámetros eléctricos y las correspondientes dimensiones del filtro. Las redes neuronales son interpoladores multidimensionales con habilidad para la aproximación de complejas funciones. Para su entrenamiento será necesario un número reducido de factores de calidad y coeficientes de acoplo. Además, al contrario que en el caso de las curvas de diseño, el proceso de cálculo de las dimensiones es inmediato y automático. No es necesaria la intervención de un diseñador que determine dichas dimensiones a partir de las curvas. Al igual que en las curvas de diseño es necesario un periodo de tiempo inicial para calcular todos los datos necesarios. En las redes neuronales, estos datos se dividen en los conjuntos de entrenamiento y validación. Si se empleasen técnicas como la IE directa o el método FEM se requeriría un tiempo excesivo para el cálculo de los datos de entrenamiento y validación. Para reducir todavía en mayor medida el periodo inicial, la nueva técnica utiliza como herramienta de análisis el método neuronal desarrollado en los Capítulos 3 y 4. Una vez entrenadas las RBFNNs, diferentes filtros multicapa apantallados cuya frecuencia central se sitúe en el ancho de banda de trabajo podrán ser diseñados rápidamente. Las salidas de las RBFNNs serán directamente las dimensiones de filtro que sintetizan una determinada función de transferencia. Es importante remarcar, que ni el periodo de cómputo de los conjuntos de entrenamiento y validación ni el tiempo de entrenamiento están incluidos en el proceso final de diseño. Cuando se diseña un filtro en la estructura multicapa en la que fueron entrenadas las redes neuronales, no es necesario ni generar un nuevo conjunto de entrenamiento ni re-entrenar las RBFNNs. Si el filtro diseñado no cumple totalmente las especificaciones, se aplica un rápido proceso de optimización basado en técnicas de gradiente. Estos métodos han sido aplicados extensamente en el diseño de circuitos de microondas como por ejemplo en el caso de circuitos de microtira en [Bandler et al., 1994a, Ureel y Zutter, 1997]. En muchas técnicas de diseño, el último paso consiste en la optimización de un filtro inicial [Alos y Guglielmi, 1997]. El método neuronal de esta sub-sección provee de una geometría inicial muy próxima al diseño óptimo. Esta característica reduce el tiempo invertido en la optimización de forma drástica. Además, si se emplea un algoritmo de gradiente se asegura al mismo tiempo que la convergencia sea la apropiada, ya que aquella depende fuertemente del punto inicial escogido. El método desarrollado no está limitado a un tipo determinado de resonador como ocurre en otras técnicas neuronales como por ejemplo en la desarrollada en [Watson et al., 1999]. Como se mostrará en los resultados, se pueden diseñar filtros formados por líneas acopladas y filtros de tipo horquilla siguiendo el mismo procedimiento.

5.3.2. Cálculo de las dimensiones del filtro a partir de los parámetros circuitales mediante redes neuronales

Como ya se ha comentado el paso más difícil en la tarea de diseño radica en la estimación de las dimensiones que sintetizan los factores de calidad y los coeficientes de acoplo. Dependiendo del tipo de resonador elegido para realizar el filtro, se pueden utilizar diferentes estructuras para calcular los mencionados parámetros eléctricos. En el cálculo del factor de calidad se opta entre una estructura compuesta por resonadores cargados de forma doble o de forma simple [Hong y Lancaster, 2001]. Así, en el caso de resonadores de líneas acopladas se puede emplear la estructura doblemente cargada de la Figura 5.17(a).

La estructura anterior posee una respuesta correspondiente al parámetro de transmisión (S_{21}) que permite la evaluación de la frecuencia de resonancia y el ancho de banda a 3 dB como

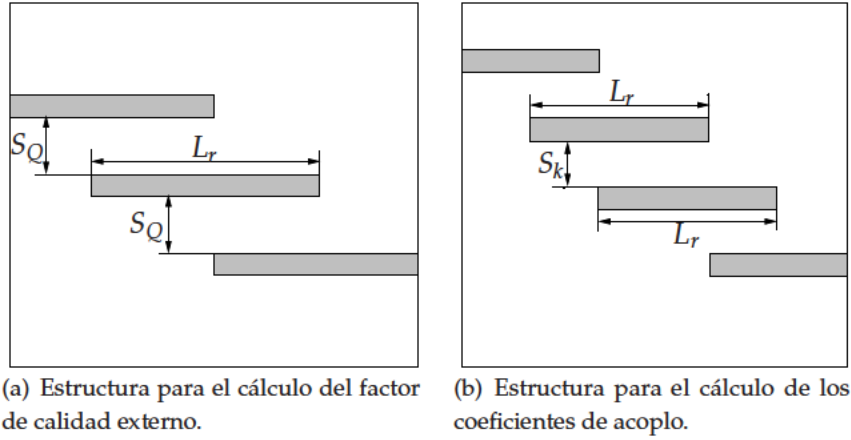


Figura 5.17: Estructuras cargadas de forma doble para el cálculo del factor de calidad externo y de los coeficientes de acoplo en resonadores de líneas acopladas.

se observa en la Figura 5.18. Utilizando estos parámetros el factor de calidad se calcula como, [Hong y Lancaster, 2001]:

$$Q_e = \frac{2f_o}{\Delta f_{3dB}} \quad (5.8)$$

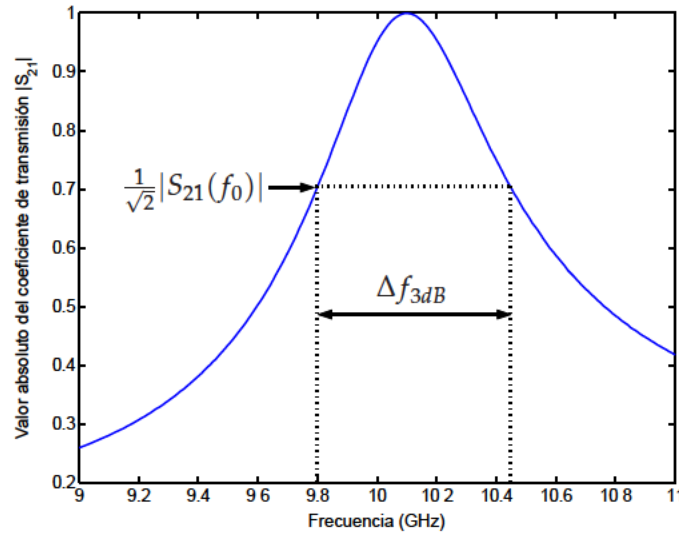


Figura 5.18: Ejemplo del módulo del coeficiente de transmisión (S_{21}) empleado para calcular el factor de calidad externo cuando se emplea una estructura cargada de forma doble como la de la Figura 5.17(a).

Para caracterizar los resonadores de tipo horquilla es más conveniente utilizar una estructura cargada de forma simple como se aprecia en la Figura 5.19(a):

Para obtener la frecuencia de resonancia y el factor de calidad se emplea la fase de la respuesta del coeficiente de reflexión (S_{11}). Así, la frecuencia de resonancia se calcula a partir del máximo de la derivada de la fase (retardo de grupo), como se observa en la Figura 5.20. El factor de calidad se

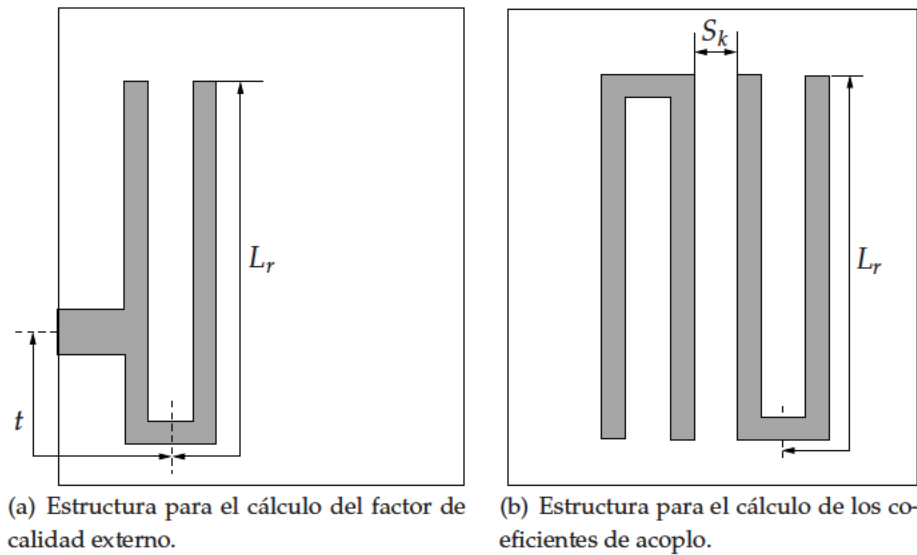


Figura 5.19: Estructuras cargadas de forma simple para el cálculo del factor de calidad externo y de los coeficientes de acoplo en resonadores de tipo horquilla.

queda determinado por aquellas frecuencias cuyas fases presentan un desfase de $\pm 90^\circ$ con respecto a la fase del coeficiente de reflexión en la frecuencia de resonancia. Con todos estos parámetros el factor de calidad se evalúa gracias a la siguiente ecuación, [Hong y Lancaster, 2001]:

$$Q_e = \frac{f_o}{\Delta f_{\pm 90^\circ}} \quad (5.9)$$

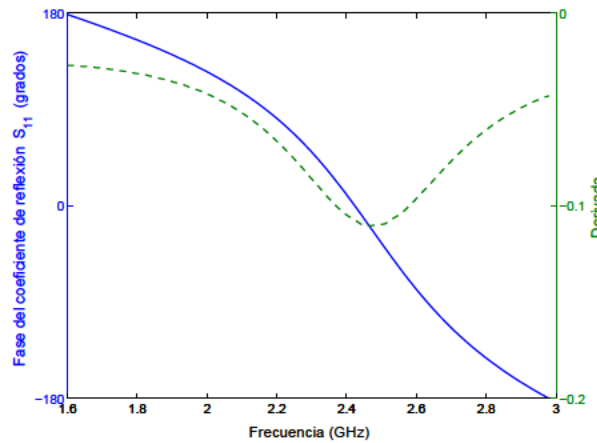


Figura 5.20: Ejemplo de fase del coeficiente de reflexión (S_{11}) (línea continua) y de su derivada (línea con guiones) empleada para calcular el factor de calidad externo y la frecuencia de resonancia cuando se utiliza una estructura cargada de forma simple como la de la Figura 5.19(a).

Los coeficientes de acoplo pueden ser calculados mediante las estructuras formadas por dos resonadores acoplados como los de las Figuras 5.17(b), para líneas acopladas, y 5.19(b), para resonadores de tipo horquilla. En estas estructuras, se obtienen dos frecuencias características f_{p1} y f_{p2} a partir

de la respuesta del coeficiente de transmisión (S_{21}), como se aprecia en la Figura 5.21. Con las frecuencias anteriores es posible calcular el coeficiente de acoplo con la expresión, [Hong y Lancaster, 2001]:

$$k = \frac{f_{p2}^2 - f_{p1}^2}{f_{p2}^2 + f_{p1}^2} \quad (5.10)$$

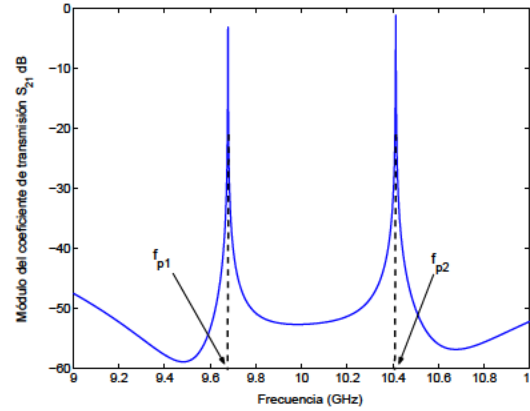


Figura 5.21: Ejemplo de módulo del coeficiente de transmisión (S_{21}) correspondiente a la estructura de la Figura 5.17(b) y de la Figura 5.19(b).

Aunque en estos ejemplos se han mostrado resonadores con forma de línea y con forma de horquilla, cualquier otro tipo de resonador puede ser empleado para calcular los factores de calidad y coeficiente de acoplo [Hong y Lancaster, 2001]. En cada caso habrá que decidir qué estructura, cargada de forma doble o simple, es más conveniente emplear para evaluar los parámetros eléctricos. En este método de diseño se han utilizado resonadores sintonizados de forma síncrona. No obstante, es posible aplicar el método a resonadores sintonizados de forma no síncrona con un nivel similar de complejidad.

Como ya se ha mencionado en la introducción, se pueden emplear curvas de diseño que relacionan las dimensiones físicas de los resonadores con los factores de calidad y los coeficientes de acoplo para hallar las dimensiones circuitales. Las redes neuronales pueden capturar este tipo de relaciones no-lineales existentes en los conjuntos de datos multidimensionales de forma eficaz. Por tanto, las redes neuronales son capaces de aproximar las curvas de diseño, y así sustituirlas, con un pequeño número de datos. Siguiendo el procedimiento de diseño neuronal inverso directo, las redes neuronales toman como entradas los parámetros eléctricos (factores de calidad, coeficientes de acoplo y frecuencia de resonancia) y ofrecen como salidas las dimensiones físicas de los resonadores que componen el circuito. Como se aprecia en la Figura 5.22 existirán en general N redes neuronales con diferentes entradas y salidas para calcular todas las dimensiones requeridas.

Cada red neuronal de la Figura 5.22 toma un conjunto particular de entradas (E) compuesto por un número determinado de parámetros eléctricos y dimensiones físicas. Cada una de las dimensiones físicas del nivel de entrada puede ser pre-fijada por el usuario o calculada por una red neuronal previa. Cada red neuronal ofrece en la salida un conjunto específico de dimensiones físicas (D) diferente de las salidas del resto de redes neuronales. Estas dimensiones físicas sintetizan los parámetros eléctricos de las correspondientes entradas de la red neuronal. Una o más salidas (D) de una

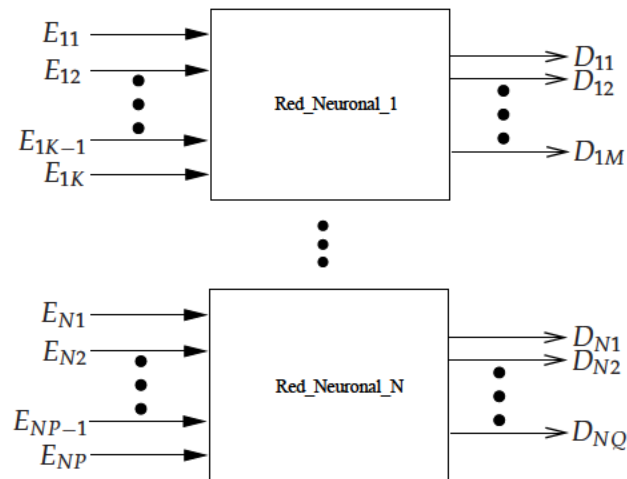


Figura 5.22: Esquema general de las redes neuronales utilizadas en la tarea de diseño. Hasta N redes neuronales pueden ser necesarias para calcular las dimensiones físicas deseadas.

red neuronal pueden ser conectadas a otra red neuronal como entradas. En el caso de resonadores de líneas acopladas son necesarias dos redes neuronales ($N = 2$). Como se expone en la Figura 5.23 la primera red neuronal calcula la relación entre la separación de acoplo (S_Q de la Figura 5.17(a)) y el factor de calidad externo. Esta red neuronal también evalúa la relación entre la longitud del resonador (L_r de la Figura 5.17(a)) y la frecuencia de resonancia (f_c). La segunda red neuronal aprende las relaciones entre la separación entre resonadores (S_k de la Figura 5.17(b)) y los coeficientes de acoplo para diferentes longitudes del resonador. Como se observa en la Figura 5.23 las dos redes neuronales se encuentran conectadas en forma de cascada. Es importante señalar que la segunda red neuronal es entrenada con diferentes longitudes de resonador, aunque cuando se utiliza para calcular la separación entre resonadores (S_k) emplea la longitud calculada con la primera red neuronal. Por esta razón, la salida de la longitud de la primera red neuronal alimenta una de las entradas de la segunda red neuronal. En ambas redes neuronales, las salidas son directamente las dimensiones de los diversos resonadores que sintetizan la función de transferencia deseada. Comparando el esquema para líneas acopladas de la Figura 5.23 con el esquema general de la Figura 5.22 se puede determinar que la primera red neuronal posee dos entradas ($K = 2$) y dos salidas ($M = 2$), mientras que la segunda red neuronal tiene dos entradas ($P = 2$) y una salida ($Q = 1$).

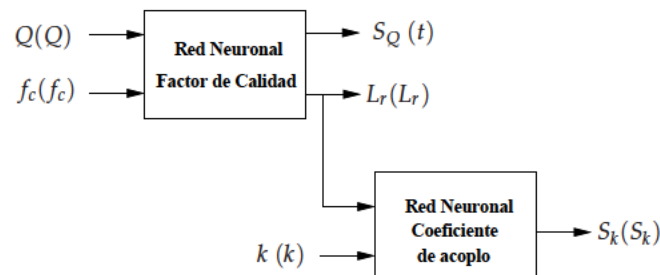


Figura 5.23: Redes neuronales cuando se emplean bien resonadores de tipo líneas acopladas bien resonadores de tipo horquilla (valores entre paréntesis).

Los filtros basados en resonadores de tipo horquilla también emplean dos redes neuronales

($N = 2$). La primera red neuronal captura las relaciones entre la longitud de "tapping" (t de la Figura 5.19(a)) y el factor de calidad externo. Además, esta red neuronal aprende la relación entre la longitud del resonador (L_r de la Figura 5.19(a)) y de la correspondiente frecuencia de resonancia (f_c). La segunda red neuronal opera en una forma similar a la segunda red neuronal descrita en el caso de los resonadores de líneas acopladas. Así, esta red neuronal calcula las relaciones entre las longitudes de los resonadores y los coeficientes de acoplo con las separaciones entre resonadores (S_k de la Figura 5.19(b)). De nuevo, ambas redes neuronales se conectan en cascada. Por lo tanto, las redes neuronales utilizadas en el problema del cómputo de las dimensiones del filtro de tipo horquilla poseen una estructura similar a las redes neuronales del filtro de líneas acopladas ($K = 2, M = 2, P = 2, Q = 1$).

La metodología empleada en los resonadores de tipo líneas acopladas y tipo horquilla es aplicable a cualquier otra clase de resonadores. El número de redes neuronales no está limitado a dos. El esquema general permite que la estructura neuronal puede ser adaptada al problema concreto para alcanzar el mejor funcionamiento. La estructura final es muy flexible ya que puede contener configuraciones en cascada como las mostradas en la Figura 5.23.

5.3.3. Método de diseño basado en Redes neuronales de funciones de base radial

Las Redes neuronales de funciones de base radial (RBFNNs) han sido seleccionadas para mostrar la capacidad de las redes neuronales para calcular las dimensiones de los resonadores. No obstante, como se ha demostrado en el Capítulo 4, las Redes neuronales de Chebychev superan el rendimiento de las RBFNNs en la aproximación de diversas funciones y también en la modelización de las funciones de Green. La elección de las RBFNNs se apoya en diversas razones. El cálculo de las dimensiones circuitales es una tarea mucho menos complicada que el problema de la aproximación de las funciones de Green. Las variaciones de la función que representa la relación entre las dimensiones físicas y los parámetros eléctricos son muchos menos abruptas que las encontradas en las funciones de Green del medio encapsulado, especialmente cerca de la fuente, ver [Hong y Lancaster, 2001]. Son necesarias muchas menos neuronas ya que el error deseado es alcanzado con cierta facilidad. El tamaño neuronal no es ahora un parámetro tan importante como en la aproximación de las funciones de Green. Tanto las RBFNNs como las Redes neuronales de Chebychev producirían redes compactas. Además, los conjuntos de entrenamiento son mucho más pequeños de modo que el método OLS podrá ser aplicado sin incurrir en ninguna saturación de la memoria del ordenador. Por tanto, en el problema tratado no es necesario utilizar las Redes neuronales de Chebychev ya que no causarían grandes beneficios respecto a las RBFNNs. Es más, éstas presentan una mayor libertad en el muestreo de la función a aproximar de modo que son preferibles en este problema de aproximación.

Para calcular los factores de calidad y los coeficientes de acoplo necesarios en los conjuntos de entrenamiento y validación se utilizó como simulador electromagnético el método neuronal desarrollado en el Capítulo 3. En concreto se ha empleado la versión del método que combina zonas cuadradas, la técnica de extracción de la singularidad mediante la multiplicación por la distancia (ρ) y el algoritmo de varianza adaptativa [Garcia et al., 2006b]. Esta versión del método es menos precisa que la que emplea la extracción de dos capas de imágenes espaciales para eliminar la singularidad. Sin embargo, la precisión alcanzada por el método neuronal es suficiente para generar los factores de calidad y coeficientes de acoplo que corresponden a la estructura bajo análisis. Este hecho se puede

comprobar gracias a la comparación de los resultados de esta sub-sección con los mostrados en la Sub-sección 5.3.5. En dicha sub-sección se exponen los factores de calidad y los coeficientes de acoplo calculados mediante el método neuronal basado en Redes neuronales de Chebychev. Este método hace uso de la técnica de extracción de dos capas de modo que se consiguen resultados un tanto más precisos sobre todo cuando las separaciones entre resonadores son reducidas. No obstante, los factores de calidad y coeficientes de acoplo calculados con los dos métodos neuronales son en general similares, de modo que conducen a filtros con respuestas muy parecidas. En este caso, la principal ventaja del método neuronal de Chebychev radica en la velocidad de análisis. Por ello, el tiempo de generación de los parámetros eléctricos será disminuido haciendo más útil y rentable el método de diseño propuesto. Si se utilizase algún método exacto de análisis electromagnético se requerirían tiempos de computación mucho más prolongados, de manera que la técnica podría resultar poco práctica.

Para mostrar el funcionamiento del método de diseño propuesto, éste se ha aplicado en dos cavidades diferentes con dos tipos distintos de resonadores. En primer lugar, se han utilizado RBFNNs para aproximar los factores de calidad y coeficientes de acoplo en la estructura de la Figura 5.27 para resonadores de tipo de líneas acopladas [Cohn, 1958]. El ancho de banda de trabajo es el mismo que se ha empleado en el desarrollo de los métodos neuronales, es decir, de 9 GHz a 11 GHz. Como se muestra en la Figura 5.23 se requieren dos redes neuronales conectadas en cascada para calcular las dimensiones del filtro. La primera RBFNN posee como entrada el factor de calidad y la frecuencia de resonancia. Las salidas son la separación de acoplo (S_Q) y la longitud del resonador (L_r). Esta red neuronal fue entrenada con 96 puntos y fue validada con un conjunto de 84 puntos diferentes. Para generar estos dos conjuntos las separaciones entre resonadores de la Figura 5.17(a) fueron muestreadas con 30 valores equidistantes entre 0,055 mm y 0,400 mm. La longitud de la línea fue muestreada con 6 valores equidistantes entre 5,45 mm y 5,95 mm. La generación completa de estos conjuntos tardó un tiempo igual a 11 horas y 33 minutos. Las muestras de validación se sitúan en posiciones intermedias respecto a las muestras de entrenamiento, tanto en el problema del factor de calidad como en el cálculo del coeficiente de acoplo. De este modo la capacidad de aproximación de las RBFNNs se comprueba de manera más exigente. El tamaño del conjunto de entrenamiento se eligió de tal forma que dicho conjunto cubriese de forma efectiva todo el espacio de trabajo. El tamaño del conjunto de validación queda determinado por el tamaño del conjunto de entrenamiento ya que se parte de un conjunto total de muestras que representan las variaciones de la función a aproximar. Una vez conseguidos dichos conjuntos, la RBFNN fue entrenada con un nivel máximo de error NMSE de 0,020. Tal como se aprecia en la Figura 5.18, para calcular el factor de calidad y la frecuencia de resonancia se debe hallar el máximo del módulo del valor absoluto del coeficiente de transmisión (S_{21}). En este trabajo, se ha utilizado la técnica denominada "golden section search" (ver [Press et al., 1997]) para localizar el máximo. Esta técnica es muy simple y no hace uso del gradiente de la función sino que emplea la comparación de los valores de la función en ciertos puntos para llevar a cabo la búsqueda del extremo.

En la Figura 5.24 se verifica el alto grado de precisión alcanzado por las RBFNNs en el cálculo de la separación de acoplo entre el resonador y los puertos de entrada y salida. Asimismo, la exactitud de la aproximación realizada en el cálculo de la longitud del resonador es elevada. En la Figura 5.25 se observa el rendimiento de la RBFNN en la obtención de la longitud del resonador. Hay que tener

en cuenta que los puntos dibujados en ambas figuras corresponden a los valores de validación o test. Se puede concluir que, ante entradas de los parámetros eléctricos (Q y f_c) distintos de los valores de entrenamiento, la primera RBFNN será capaz de devolver las dimensiones físicas apropiadas.

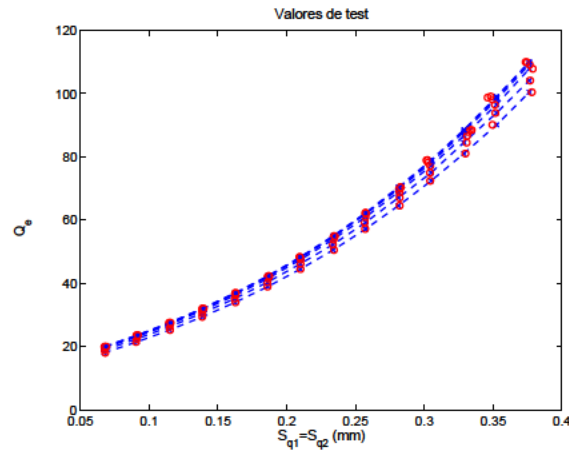


Figura 5.24: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.17(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

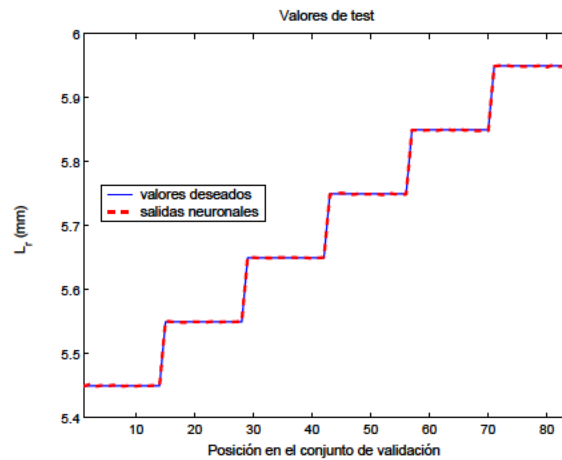


Figura 5.25: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.17(a).

La segunda RBFNN evalúa las separaciones entre las líneas acopladas (S_k) a partir de la longitud calculada con la primera RBFNN y el coeficiente de acoplo deseado. Los conjuntos de entrenamiento y validación fueron generados por medio de un proceso similar al aplicado en la primera RBFNN. Así, la separación de acoplo fue muestreada con 23 valores equidistantes entre 0,30 mm y 1,90 mm. La longitud del resonador fue obligatoriamente muestreada entre los mismos límites utilizados en el problema del factor de calidad (5,45-5,95 mm). Además, también se utilizaron 6 valores equidistantes.

El conjunto de entrenamiento se compuso con 72 puntos mientras que el conjunto de validación constó de 66 puntos diferentes a los de entrenamiento. El tiempo total invertido en la producción de todos los puntos necesarios fue de 10 horas y 31 minutos. Al igual que en el problema del factor de calidad, para localizar las frecuencias (f_{p1}, f_{p2}) se empleó la técnica de "golden section search". El error NMSE máximo también se fijó en un valor de 0,020. Este bajo nivel de error asegura que las salidas de la segunda RBFNN sigan con fidelidad la evolución de los coeficientes de acoplo como se ve en la Figura 5.26.

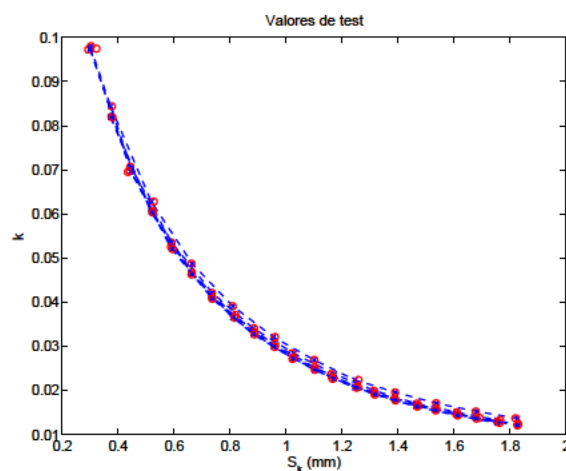


Figura 5.26: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.17(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

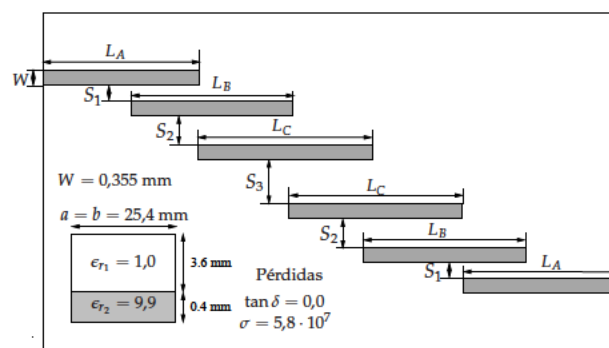


Figura 5.27: Estructura multicapa apantallada (coincide con la estructura mostrada en la Figura 3.38) y filtro de cuarto orden.

Las RBFNNs conectadas en forma de cascada también fueron empleadas para calcular las dimensiones de los filtros cuando éstos están compuestos por resonadores de tipo horquilla. El ancho de banda de trabajo es 1 – 3 GHz. El método se aplicó en la estructura dibujada en la Figura 5.28. Siguiendo el modelo descrito en la Figura 5.23 la primera red neuronal toma como entradas la distancia "tapping" (t) y la longitud del resonador (L_r). Para representar correctamente el espacio de entrada y salida de esta red neuronal fueron necesarios hasta 200 puntos. El conjunto de entrenamiento fue

dotado con 110 puntos. El conjunto usado para probar la capacidad de generalización de la RBFNN está compuesto por 90 puntos. Para generar estos conjuntos de entrenamiento y validación, la distancia de "tapping" fue muestreada con 20 valores equidistantes entre 4,5 mm y 10,5 mm. La longitud del resonador fue muestreada con 10 valores equidistantes entre 16,4 mm y 24,4 mm. El tiempo invertido en la obtención de todos estos valores fue de 24 horas y 43 minutos. El error NMSE máximo se fijó de nuevo en 0,020. En la Figura 5.29 se comprueba la alta precisión conseguida por la primera RBFNN en el cómputo de las distancias de "tapping" (t). Por otra parte, en la Figura 5.30 se muestran los valores de salida de la primera RBFNN correspondientes a la longitud del resonador. Al igual que en el caso anterior, la red neuronal alcanza una gran precisión.

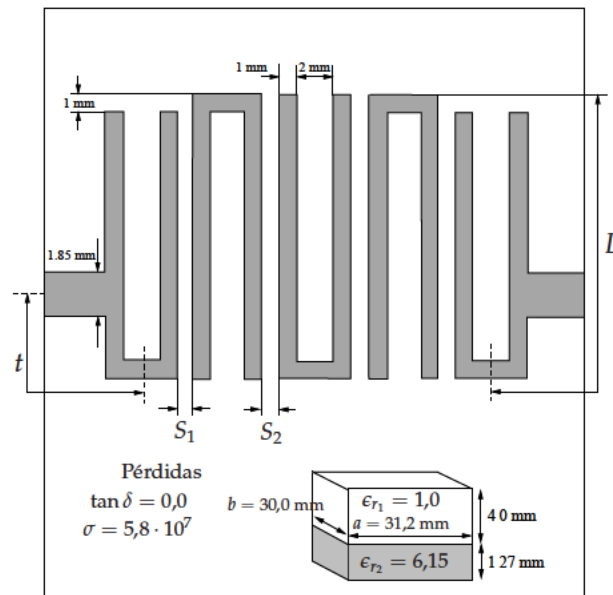


Figura 5.28: Estructura multicapa apantallada (coincide con la estructura mostrada en la Figura 3.69) y filtro de cuarto orden.

La segunda RBFNN calcula las separaciones entre resonadores (S_k) de modo similar a la segunda red neuronal del ejemplo previo. Esta red neuronal toma la longitud del resonador evaluada con la primera RBFNN y el coeficiente de acoplo requerido. El espacio de entrada fue muestreado con 150 puntos para representar las variaciones de la separación entre resonadores. Para ello, dicha separación fue muestreada con 15 valores equidistantes entre 0,20 mm y 1,00 mm. La longitud fue asimismo muestreada con 10 valores equidistantes entre 16,4 mm y 24,4 mm. El tiempo total de generación fue de 13 horas y 33 minutos. El error máximo NMSE se fijó en 0,020. De nuevo se utilizó, tanto en el cálculo del coeficiente de acoplo como en el del factor de calidad, el método de "golden section search" para calcular los extremos. En la Figura 5.31 se demuestra el alto grado de precisión conseguido en el cálculo de la separación entre resonadores.

Para todos los ejemplos vistos, el tiempo de entrenamiento de las RBFNNs es tan solo de unos pocos segundos. Esto es debido al reducido número de muestras de entrenamiento requeridas para representar adecuadamente las relaciones entre los parámetros eléctricos y las dimensiones de los resonadores. En cambio, el tiempo invertido en la generación de los conjuntos de entrenamiento y validación es muy largo en comparación con el periodo de entrenamiento. No obstante, al igual que el tiempo de entrenamiento, este tiempo es un periodo invertido antes de la fase real de diseño.

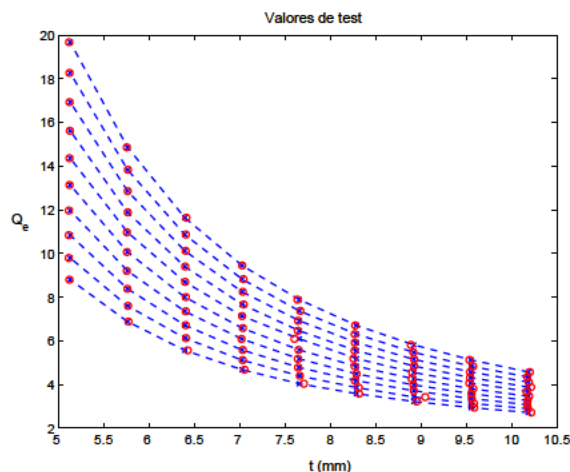


Figura 5.29: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.19(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

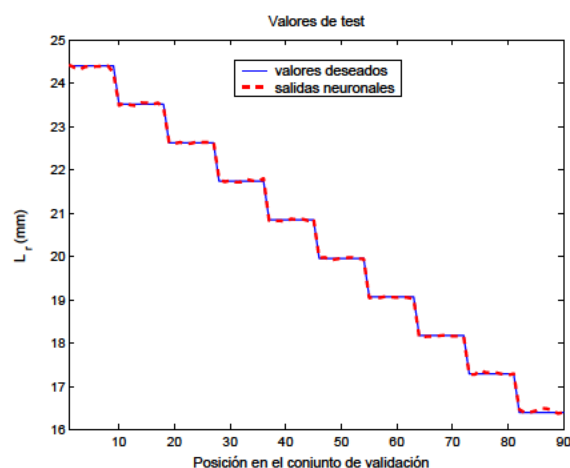


Figura 5.30: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.19(a).

No pertenece, por tanto, a la fase real de diseño. Una vez entrenadas, las RBFNNs inter-conectadas calculan las dimensiones del filtro en fracciones de segundo. Si fuera necesaria una fase final de optimización, el filtro calculado con las redes neuronales constituirá un punto inicial muy cercano al óptimo. Este hecho posibilita que los tiempos de diseño sean breves, en comparación con la situación en la que un filtro inicial de calidad no se encuentra disponible. En tales ocasiones, las técnicas basadas en el uso del gradiente pueden no convergir a una solución satisfactoria. En estos casos, una alternativa consiste en la aplicación de una técnica de optimización como la de los Algoritmos genéticos. Como vimos en la Sección anterior el AG, incluso empleando funciones de rendimiento apropiadas,

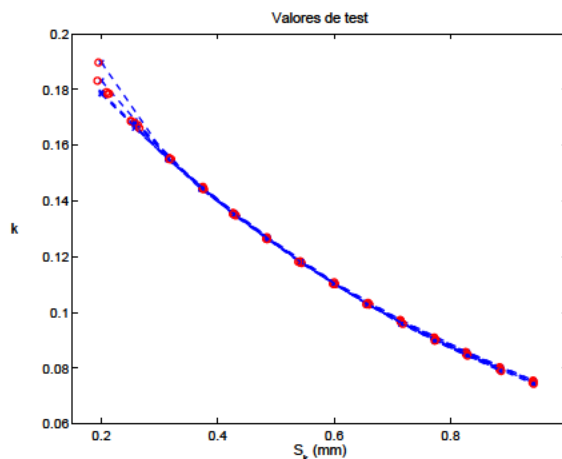


Figura 5.31: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.19(b). Los valores de validación corresponden a la línea continua con cruces. Las salidas de la RBFNN corresponden a los círculos.

requerirá un prolongado tiempo de ejecución para encontrar la solución. Este tiempo puede llegar a ser muy superior al tiempo invertido en la generación de los conjuntos de entrenamiento y validación de las RBFNNs usados en el presente método.

5.3.4. Ejemplos de diseño de filtros apantallados con el método de diseño basado en Redes neuronales de funciones de base radial: simulador electromagnético basado en Redes neuronales de funciones de base radial

En esta sub-sección se presentan diversos ejemplos que prueban la capacidad de las RBFNNs entrenadas en las dos cavidades utilizadas. En la cavidad de la Figura 5.27 se han diseñado dos filtros de líneas acopladas. Los límites tomados en los conjuntos de entrenamiento de las RBFNNs en esta estructura permiten sintetizar diferentes funciones de transferencia con frecuencia central entre 9,50 y 10,25 GHz. El primer filtro tiene un ancho de banda relativo del 5,0 % centrado en 10,25 GHz, cerca por tanto del límite del conjunto de entrenamiento. De este modo se pone a prueba la capacidad de aproximación de las RBFNNs en aquella parte del conjunto de entrenamiento en la que se pueden dar los mayores errores. Esto es debido a la falta de datos más allá de los límites del conjunto de entrenamiento. El rizado de este filtro es de 0,10 dB, la atenuación en la banda de rechazo debe ser mejor que 20 dB en el rango de frecuencias $f \leq 9,70$ GHz y $f \geq 10,80$ GHz. Con estas especificaciones, si la función de transferencia elegida es de tipo Chebychev, entonces es necesario emplear un filtro de cuarto orden [Matthaei et al., 1980]. Las longitudes y separaciones calculadas con las RBFNNs de la sub-sección previa se exponen en la segunda y tercera columnas de la Tabla 5.4.

Estas dimensiones corresponden al diseño del filtro inicial, con la respuesta mostrada en la Figura 5.32. Inicialmente, todas las líneas acopladas poseen la misma longitud ya que en este tipo de filtros la separación entre líneas afecta poco a la frecuencia de resonancia del resonador. Los pará-

Tabla 5.4: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27.

Parámetro	Sep. inicial en mm	Long. inicial en mm	Sep. óptima en mm.	Long. óptima en mm.
$Q_e = 22,72$	0.0992	5.4541	0.1326	5.4650
$k_{12} = 0,0405$	0.7855	5.4541	0.7748	5.4625
$k_{23} = 0,0321$	0.9959	5.4541	0.9941	5.4625
$k_{34} = 0,0405$	0.7855	5.4541	0.7748	5.4650

metros S del filtro inicial cumplen prácticamente todas las especificaciones. Para que el filtro cumpla completamente las especificaciones es necesario aplicar un proceso de optimización. Esta fase final de ajuste se aplica a todas las separaciones y longitudes de los resonadores. Por tanto, ahora se obtendrán resonadores con diferentes longitudes. El método de optimización empleado está basado en técnicas que usan el gradiente de la función a optimizar. En particular se trata del algoritmo minimax descrito en [Bandler et al., 1985]. Después de tan solo dos iteraciones del mencionado algoritmo, la respuesta del filtro es mejorada hasta el punto de satisfacer totalmente la máscara de atenuación como se observa en la Figura 5.32. Las dimensiones del filtro optimizado se muestran en la Tabla 5.4. Las diferencias entre los valores iniciales y los valores finales son muy pequeñas, del orden del 2,0 %. Tan solo en la separación entre los puertos de entrada y salida y los resonadores correspondientes se aprecia una diferencia mucho mayor entre el valor inicial y el final (en torno al 25,0 %). Aunque el filtro inicial no cumple todas las especificaciones, este filtro constituye un punto inicial muy adecuado para aplicar un sencillo proceso de optimización como se deduce de su respuesta y dimensiones. Así, las rápidas técnicas de optimización basadas en el gradiente pueden ser aplicadas con excelentes resultados. Los resultados del filtro inicial y del filtro optimizado se obtienen con el método neuronal desarrollado en [García et al., 2006b]. Para validar los resultados del filtro optimizado se simuló su respuesta con la técnica de la IE utilizando las funciones de Green evaluadas numéricamente y con el programa comercial ADS[®]. En la Figura 5.33 se observa que existe una apreciable similitud entre todos los métodos utilizados.

El segundo filtro de líneas acopladas diseñado es un filtro de segundo orden con frecuencia central 9,85 GHz. El ancho de banda relativo se fijó en un 1,0 %, el rizado es de 0,10 dB. La atenuación en la banda de rechazo debe superar un umbral de 15 dB en las bandas $f \leq 9,60$ y $f \geq 10,10$ GHz. Para generar este filtro se eligió de nuevo una función de transferencia de tipo Chebychev. Con los valores de los elementos del prototipo paso bajo se hallan los factores de calidad y coeficientes de acoplo que, insertados como entradas en las RBFNNs junto a la frecuencia de resonancia, producen las dimensiones que se desglosan en la Tabla 5.5. Del mismo modo que en el filtro de cuarto orden, todos los resonadores poseen inicialmente las mismas longitudes.

La respuesta del filtro inicial se muestra en la Figura 5.34. A pesar de que la respuesta cumple prácticamente todas las especificaciones, es necesario de nuevo aplicar una fase de optimización. El algoritmo minimax requiere en este caso de nueve iteraciones para conseguir una respuesta que cumpla las especificaciones sin excepciones, ver Figura 5.34. Las nuevas dimensiones representan

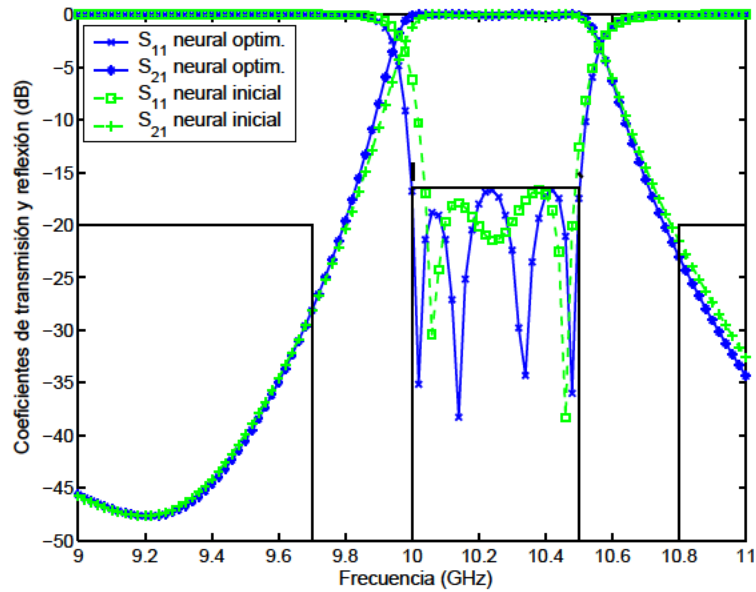


Figura 5.32: Solución inicial y optimizada del filtro de cuarto orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Se muestra la máscara de atenuación con líneas continuas.

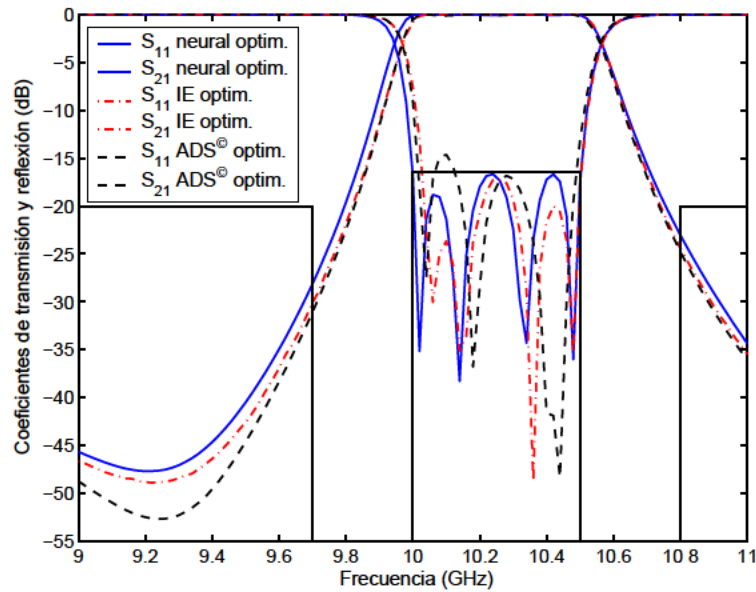


Figura 5.33: Solución optimizada del filtro de cuarto orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral y la simulación con el programa ADS[®]. Se muestra la máscara de atenuación con líneas continuas.

ligeras alteraciones respecto a los valores iniciales como se ve en la Tabla 5.5. Por lo tanto, el filtro inicial es un excelente punto de partida para el proceso de optimización. La validación de la respuesta optimizada se llevó a cabo con la solución directa de la IE y el programa comercial ADS[®]. De nuevo, se observa un gran parecido entre las respuestas de todos los métodos empleados.

Tabla 5.5: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de segundo orden de líneas acopladas de la caja de la Figura 5.27.

Parámetro	Sep. inicial en mm	Long. inicial en mm	Sep. óptima en mm.	Long. óptima en mm.
$Q_e = 83,034$	0.3180	5.7054	0.3334	5.7036
$k_{12} = 0,0140$	1.6641	5.7054	1.6280	5.7036

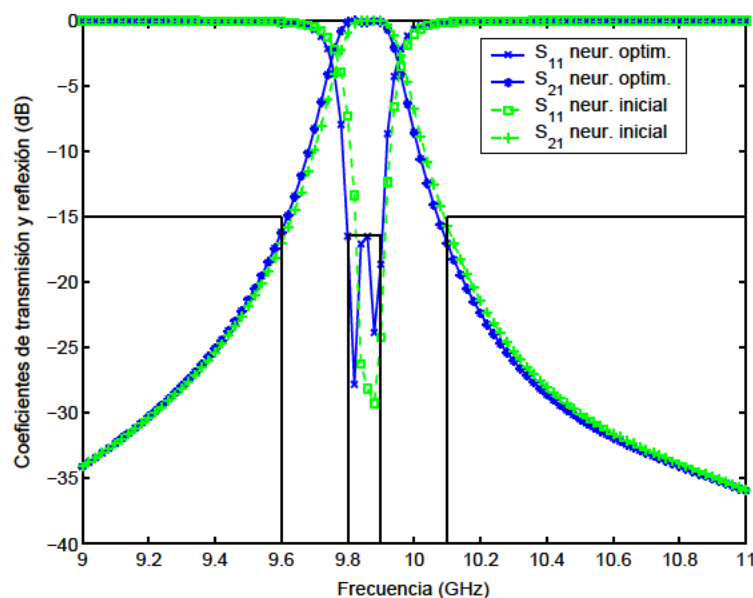


Figura 5.34: Solución inicial y optimizada del filtro de segundo orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Se muestra la máscara de atenuación con líneas continuas.

Finalmente, para probar la capacidad de las RBFNNs en el diseño de filtros en la cavidad dibujada en la Figura 5.28 se generó un filtro de tipo horquilla. Como se observa en la Figura 5.23 las RBFNNs han sido entrenadas en esta cavidad para entregar las dimensiones de filtros formados por resonadores de tipo horquilla. Con el rango de variación de la longitud de resonador (L_r), empleado en el entrenamiento de las dos RBFNNs, se pueden diseñar filtros con diferentes funciones de transferencia y con una frecuencia central entre 1,60 GHz y 2,55 GHz. El filtro diseñado respeta estos límites ya que su frecuencia central es 2,30 GHz. El ancho de banda relativo es del 20 %. Como en los filtros de líneas acopladas se escogió una función de transferencia de tipo Chebychev. El rizado en la banda de paso se fijó en 0,1 dB. La atenuación de las bandas rechazadas debe ser mejor que 25 dB en los rangos $f \leq 1,90$ GHz y $f \geq 2,75$ GHz. Para satisfacer las atenuaciones deseadas en las bandas de paso y rechazo es necesario que el filtro sea de orden quinto [Matthaei et al., 1980]. En la Tabla 5.6 se exponen las dimensiones calculadas con las RBFNNs para la longitud de "tapping" y las separaciones y longitudes de los resonadores. Todos los resonadores poseen la misma longitud inicial ya que, de nuevo, las separaciones entre resonadores apenas afectan a la frecuencia de resonancia de los resonadores. La respuesta simulada con el método neuronal para el filtro sintetizado con las

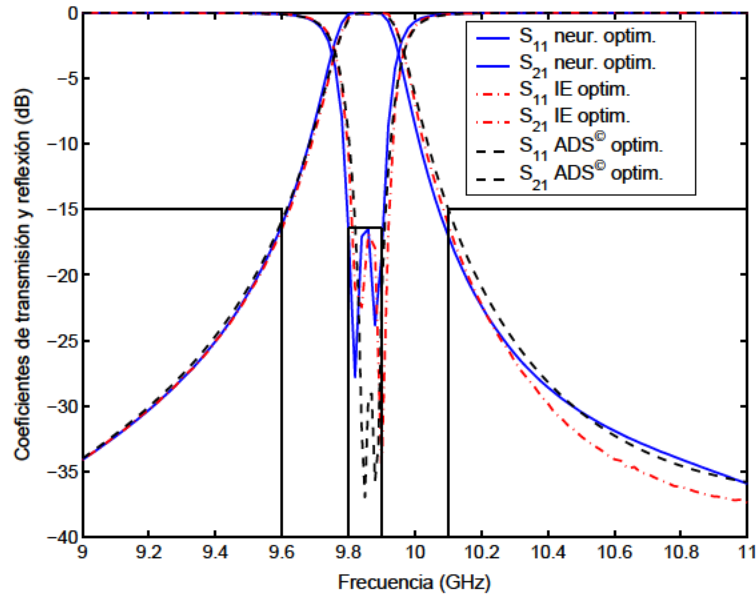


Figura 5.35: Solución optimizada del filtro de segundo orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral y la simulación con el programa ADS[®]. Se muestra la máscara de atenuación con líneas continuas.

dimensiones mencionadas de muestra en la Figura 5.36.

Tabla 5.6: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de quinto orden de tipo horquilla de la caja de la Figura 5.28.

Parámetro	Tap. y sep. iniciales en mm	Long. inicial en mm	Tap. y sep. óptimas en mm.	Long. óptima en mm.
$Q_e = 5,755$	6.770	18.05	6.6302	18.147
$k_{12} = 0,1589$	0.2900	18.05	0.3116	18.147
$k_{23} = 0,1211$	0.5190	18.05	0.4696	18.147
$k_{34} = 0,1211$	0.5190	18.05	0.4633	18.147
$k_{45} = 0,1589$	0.2900	18.05	0.3075	18.147

Como ya se hizo en los filtros de líneas acopladas se debe aplicar una fase de optimización para que el filtro diseñado satisfaga la máscara de atenuación. Tras nueve iteraciones del algoritmo minimax se obtiene la respuesta optimizada dibujada en la Figura 5.36. Ahora, tanto el coeficiente de transmisión como el coeficiente de reflexión cumplen las especificaciones. En este caso, para mantener un número limitado de variables de optimización, todos los resonadores fueron forzados a mantener la misma longitud. Las dimensiones del filtro final exhiben pequeñas diferencias respecto de las dimensiones iniciales (entre el 2 % y el 12 %). Este hecho, unido al grado de cumplimiento de las especificaciones, permite aseverar que el filtro inicial es un excelente punto de inicio para el proceso de optimización. Dicho proceso debe corregir la respuesta inicial solo ligeramente.

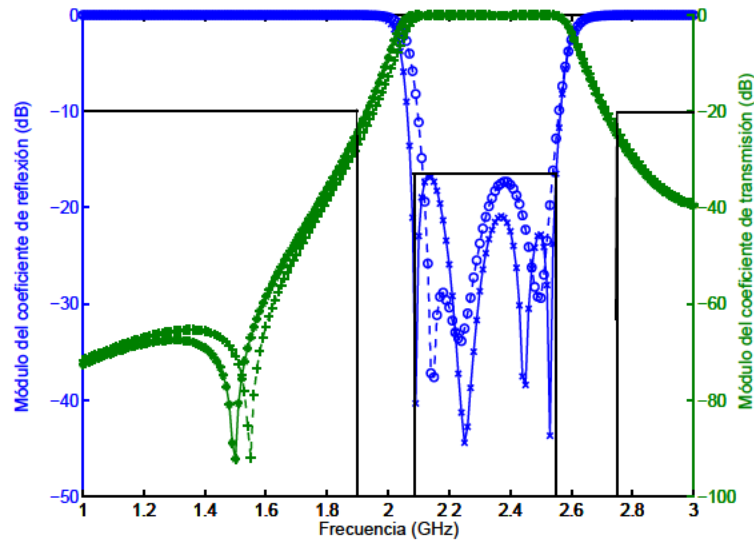


Figura 5.36: Solución inicial (línea con guión y puntos) y optimizada (línea continua) del filtro de quinto orden de tipo horquilla correspondiente a la caja de la Figura 5.28. Se muestra la máscara de atenuación con líneas continuas.

Para verificar la técnica desarrollada y la respuesta obtenida con el método neuronal se fabricó el filtro de tipo horquilla de cinco polos, como se aprecia en la Figura 5.37. Los resultados incluidos en la Figura 5.38 son muy similares a las mediciones efectuadas y las simulaciones correspondientes a la solución directa. El prototipo fabricado exhibe una pérdidas mínimas de inserción dentro de la banda de paso iguales a $-1,79$ dB.

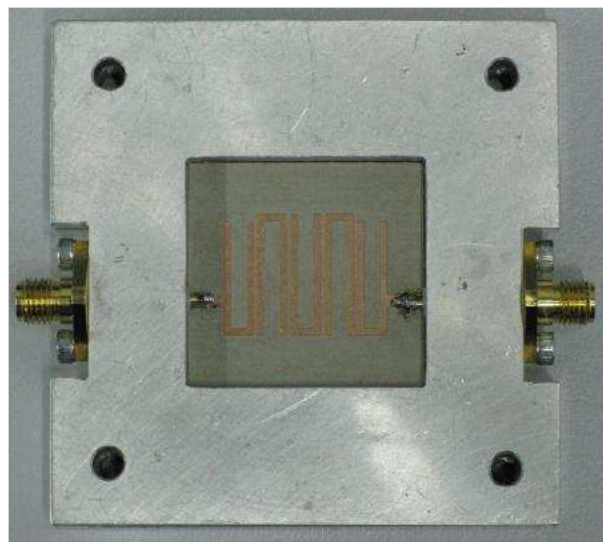


Figura 5.37: Fotografía del filtro de cinco polos de tipo horquilla fabricado.

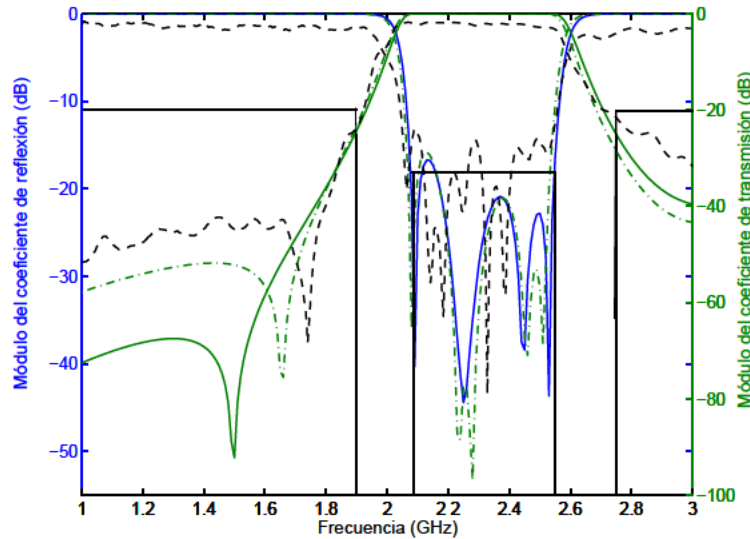


Figura 5.38: Solución optimizada (línea continua) del filtro de quinto orden de tipo horquilla correspondiente a la Figura 5.28. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral (línea con guión y puntos) y las medidas (línea con guiones). Se muestra la máscara de atenuación con líneas continuas.

5.3.5. Ejemplos de diseño de filtros apantallados con el método de diseño basado en Redes neuronales de funciones de base radial: simulador electromagnético basado en Redes neuronales de Chebychev

En el Capítulo 4 se desarrolló el método neuronal de análisis de circuitos multicapa apantallados basado en las Redes neuronales de Chebychev. Gracias al uso eficiente de la técnica de entrenamiento OLS, las Redes neuronales de Chebychev consiguen aproximar las funciones de Green con un número de neuronas mucho más reducido que las RBFNNs y con una precisión similar. Por tanto, si se utiliza como simulador electromagnético el método neuronal basado en Redes neuronales de Chebychev, en lugar del método fundamentado en RBFNNs, los tiempos de generación de los conjuntos de entrenamiento y validación en el problema de diseño serán disminuidos en gran medida. Esto convertirá en mucho más rentable y atractivo el nuevo método de diseño. Una vez obtenidos todos los datos que representan las relaciones entre las dimensiones físicas y los parámetros eléctricos, se utilizará un conjunto de RBFNNs para capturar dichas relaciones al igual que en la sub-sección anterior. No es necesario emplear Redes neuronales de Chebychev en esta tarea ya que el conjunto de entrenamiento es muy pequeño. Además, la labor de aproximación no resulta complicada debido a la ausencia de variaciones bruscas. Por todo ello, el número de neuronas de las RBFNNs en este problema de aproximación es muy pequeño, de modo que si se empleasen las Redes neuronales de Chebychev no se conseguiría una gran ganancia en el tiempo de generación de las dimensiones. Las RBFNNs tan solo cuentan con un número máximo de veinte neuronas de forma que el cálculo de las dimensiones de los resonadores es instantáneo. Las RBFNNs conectadas en cascada serán re-entrenadas con los datos generados con el método de análisis basado en Redes neuronales de Chebychev.

Los tiempos de generación de los conjuntos de entrenamiento y validación son disminuidos

considerablemente. En el caso de los resonadores de líneas acopladas, la obtención de un total de 150 datos requiere un tiempo de 2 horas y 49 minutos. De este conjunto de datos se extraen 78 valores para el entrenamiento de las RBFNNs y 72 valores para la validación. El tiempo mencionado es muy inferior por tanto al tiempo necesitado en el caso de utilizar el método neuronal de análisis basado en RBFNNs (11 horas y 33 minutos). Para entrenar y validar la segunda RBFNN se emplean asimismo un total de 150 datos. El conjunto de entrenamiento está compuesto de nuevo por 78 datos mientras que el resto pertenece al conjunto de validación. El periodo invertido para calcular los coeficientes de acoplo en este caso asciende a 4 horas y 40 minutos. Menos de la mitad del tiempo necesitado con el método neuronal basado en RBFNNs (10 horas y 31 minutos). En las Figuras 5.39 y 5.40 se observan los conjuntos de validación obtenidos en el cálculo de la separación entre puerto y resonador y en el cálculo de la longitud del resonador respectivamente. Los valores son muy similares a los conseguidos con el método de análisis fundamentado en RBFNNs (ver Figuras 5.24 y 5.25). La precisión de la primera de las dos RBFNNs conectadas en cascada es elevada, al igual que en la sub-sección anterior. Los datos de validación del problema del cálculo de las relaciones entre coeficientes de acoplo y separaciones entre resonadores se exponen en la Figura 5.41. La precisión de la segunda RBFNN es lo suficientemente alta para asegurar que ante un nuevo par de datos de entrada se devolverá la separación adecuada.

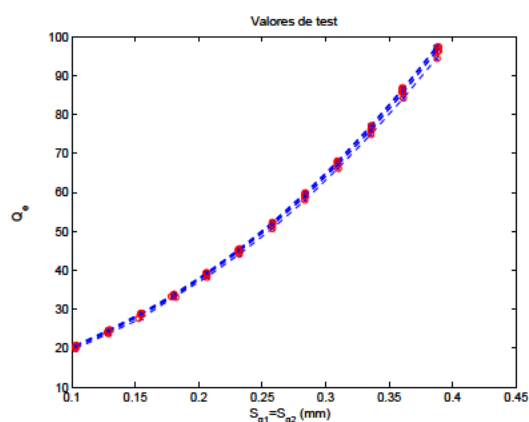


Figura 5.39: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de líneas acopladas. Los factores de calidad han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.17(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

Para probar que las RBFNNs entrenadas con los nuevos datos de entrenamiento son aptas para la tarea de diseño se volvió a calcular las dimensiones de los dos filtros de líneas acopladas de la sub-sección anterior. Las nuevas dimensiones se exhiben en la Tabla 5.7 para el filtro de cuarto orden y en la Tabla 5.8 para el filtro de segundo orden. En general, todas las longitudes y separaciones son similares a las obtenidas con los datos de entrenamiento generados con el método de análisis basado en RBFNNs. En la Figura 5.42 se exhibe la respuesta del filtro de cuarto orden inicial obtenida con dichas dimensiones. Es significativo el valor de separación entre los puertos y los resonadores calculada para el filtro de cuarto orden. Como se observa en la Tabla 5.7 es un valor inicial (0,1209 mm)

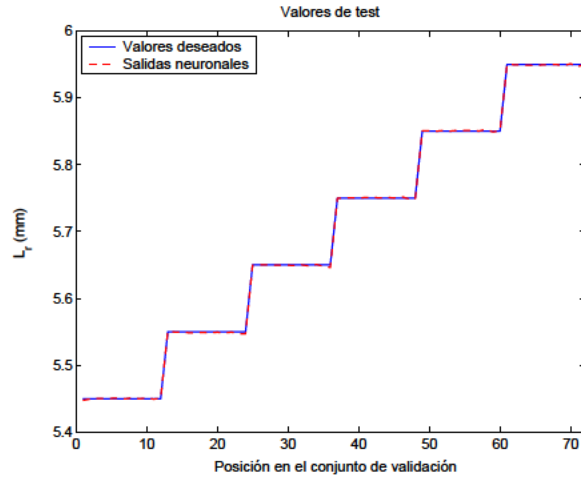


Figura 5.40: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de líneas acopladas. Las frecuencias de resonancia han sido calculadas mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.17(a).

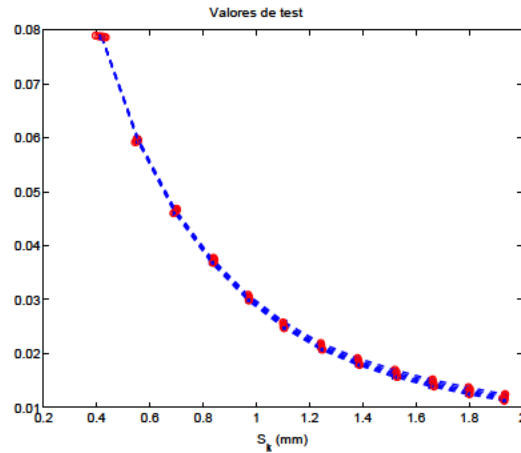


Figura 5.41: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de líneas acopladas. Los coeficientes de acoplo han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.17(b). Los valores de validación corresponden a la línea continua con cruces. Las salidas de la RBFNN corresponden a los círculos.

mucho más cercano al valor correspondiente al filtro optimizado (0,1326 mm) que el valor inicial conseguido con las RBFNNs de la Sub-sección anterior (0,0992 mm, ver Tabla 5.4). El método neuronal de análisis empleado en la sub-sección previa utiliza la técnica de multiplicación por (ρ) para eliminar la singularidad. Por ello, cuando la separación entre tiras metálicas es menor, el método de análisis resulta menos preciso, de modo que los valores de entrenamiento expresarán de forma menos real-

ista las relaciones entre las dimensiones físicas y los parámetros eléctricos en esa parte del espacio de solución. En cambio, el método de análisis basado en Redes neuronales de Chebychev utiliza la técnica de extracción de dos capas para suprimir la singularidad. Este procedimiento es más preciso de forma que los valores de entrenamiento representan más acertadamente la relación entre entrada y salida.

La respuesta mostrada en la Figura 5.42 satisface en gran medida la máscara de atenuación. Sin embargo, es necesario aplicar un proceso de optimización para que se cumplan las especificaciones en todo el ancho de las bandas de paso y rechazo. En este caso, en lugar de aplicar un algoritmo basado en el uso del gradiente de una función objetivo hemos optado por alterar de manera directa y paso a paso las dimensiones del filtro. Así, el diseñador debe cambiar las separaciones y longitud de los resonadores para conseguir el cumplimiento total de las especificaciones. Con este procedimiento se pretende demostrar, por un lado la facilidad para optimizar el filtro inicial y por otro lado que la utilización en el futuro de técnicas de optimización de lógica difusa es factible. La lógica difusa intenta emular la forma en que los humanos resolvemos problemas mediante reglas que contienen información lingüística. Así, estas técnicas se basan en la aplicación de reglas del tipo *si... entonces* sobre conjuntos difusos [Mendel, 1995]. En el diseño de los filtros basados en resonadores existen reglas bien conocidas que pueden ser implantadas en un optimizador de lógica difusa. Por ejemplo, la separación entre los puertos de entrada y salida y los resonadores adyacentes controla el ancho de banda relativo de un filtro paso banda. Las separaciones entre los resonadores influyen en la atenuación de la banda de paso. En todos los ejemplos siguientes la optimización del filtro inicial se ha llevado a cabo mediante la aplicación de estas reglas. La optimización directa de los filtros diseñados en esta sub-sección no resulta una tarea pesada gracias a la rapidez del método de análisis basado en Redes neuronales de Chebychev. Como se mostrará a continuación los resultados son satisfactorios.

Tabla 5.7: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial.
Filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27.

Parámetro	Sep. inicial en mm	Long. inicial en mm
$Q_e = 22,72$	0.1209	5.4785
$k_{12} = 0,0405$	0.7823	5.4785
$k_{23} = 0,0321$	0.9406	5.4785
$k_{34} = 0,0405$	0.7823	5.4785

Tabla 5.8: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial.
Filtro de segundo orden de líneas acopladas de la caja de la Figura 5.27.

Parámetro	Sep. inicial en mm	Long. inicial en mm
$Q_e = 83,034$	0.3505	5.7194
$k_{12} = 0,0140$	1.6914	5.7194

El proceso de optimización directa se aplicó al filtro de cuarto orden de líneas acopladas. Las

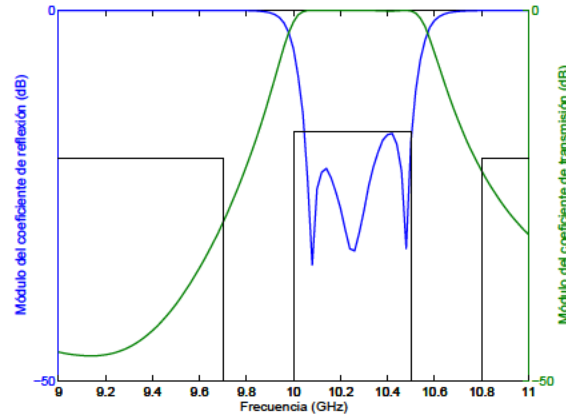


Figura 5.42: Respuesta del filtro inicial de cuarto orden de líneas acopladas correspondiente a las dimensiones de la Tabla 5.7. Asimismo, se muestra la máscara de atenuación.

variables de optimización son cinco: las tres separaciones denominadas S_1 , S_2 y S_3 y las dos longitudes L_B y L_C de la Figura 5.27. En la Tabla 5.9 se exponen los incrementos aplicados sobre dichas separaciones y longitudes. Son necesarias diez iteraciones para alcanzar la respuesta que cumple las especificaciones mostrada en la Figura 5.44. El proceso se inicia con el ajuste de las longitudes de los resonadores para equilibrar la atenuación en los lóbulos externos de la banda de paso. Existen algunos incrementos que provocan un empobrecimiento de la respuesta del filtro como por ejemplo en la iteración tercera, (ver Figura 5.43(b)). El ajuste de las separaciones permite mejorar las prestaciones del filtro hasta conseguir una respuesta que prácticamente cumple las especificaciones en todas las frecuencias como la de la Figura 5.43(d) correspondiente a la quinta iteración. Es difícil que un algoritmo de gradiente, por rápido que sea, consiga optimizar el filtro con tan solo diez simulaciones de la respuesta. Hay que tener en cuenta que el cálculo del gradiente de la función de error puede requerir dos simulaciones de la respuesta circuital por cada variable de optimización [Nocedal y Wright, 1999]. La principal ventaja del algoritmo de gradiente en este caso radica en la automatización del proceso. Sin embargo, la aplicación de las reglas que permiten la fácil optimización directa también puede ser automatizada mediante la lógica difusa.

Tabla 5.9: Incrementos en mm. aplicados a las separaciones y longitudes iniciales en la optimización el filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27.

Variable	Iteración									
	1	2	3	4	5	6	7	8	9	10
ΔS_1	0.000	0.000	0.000	0.025	0.025	0.027	0.0270	0.0290	0.0320	0.0320
ΔS_2	0.000	0.000	0.000	0.000	0.000	0.000	0.0000	0.0000	0.0050	0.0075
ΔS_3	0.000	0.000	0.000	0.000	0.010	0.010	0.0100	0.0100	0.0100	0.0100
ΔL_B	0.015	0.015	0.015	0.015	0.015	0.015	0.0125	0.0125	0.0125	0.0125
ΔL_C	0.015	0.020	0.025	0.020	0.020	0.020	0.0150	0.0150	0.0150	0.0150

Hasta el momento todos los filtros diseñados sintetizan funciones de transferencia de tipo Chebychev. No obstante, el método desarrollado sirve para sintetizar otros tipos de respuesta. Con el objetivo de demostrar la adaptabilidad de la metodología se han diseñado diversos filtros de tipo

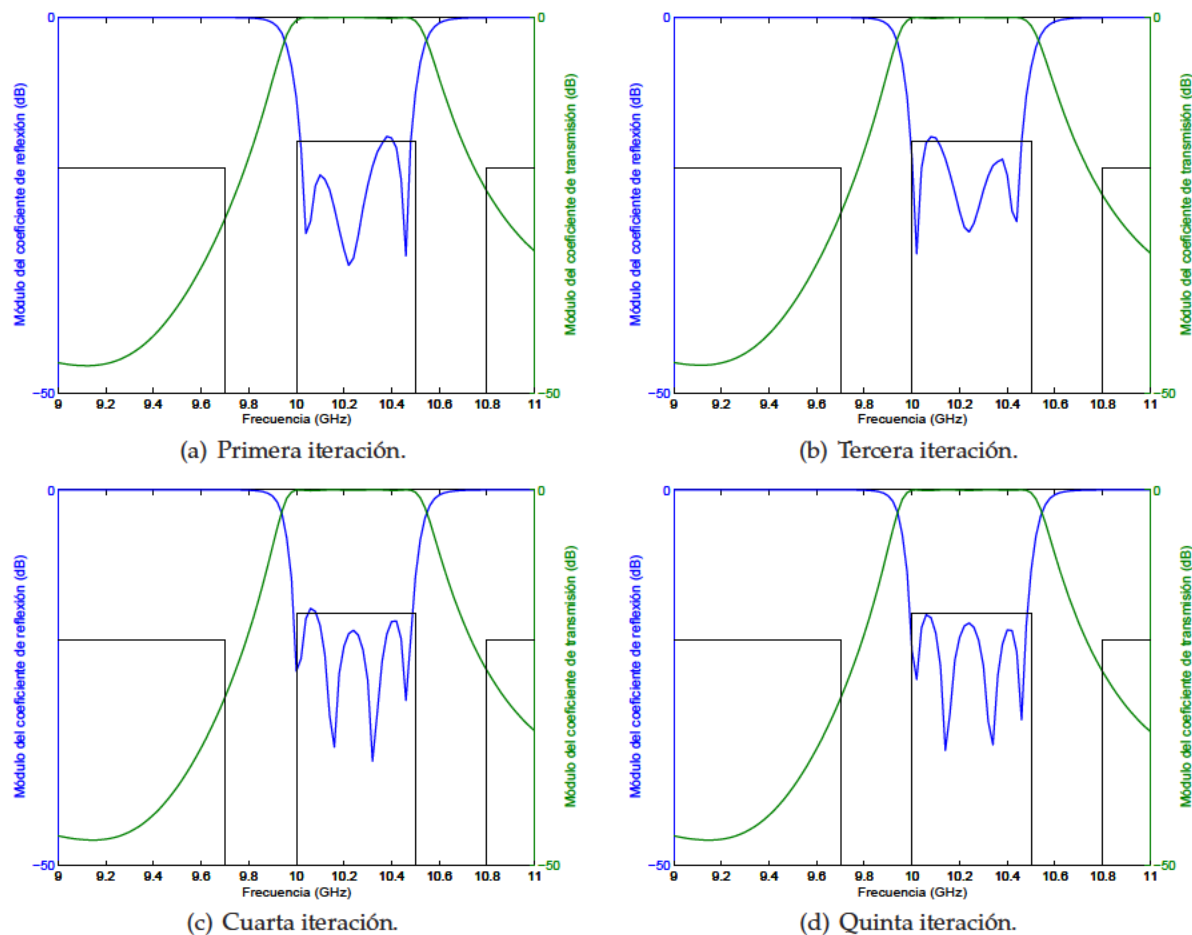


Figura 5.43: Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de cuarto orden de tipo Chebyshev realizado con líneas acopladas.

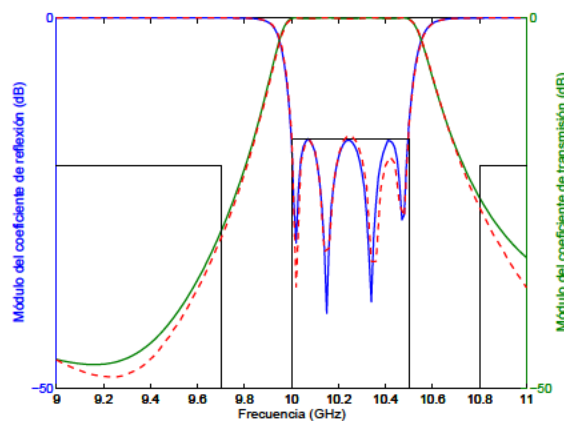


Figura 5.44: Respuesta del filtro optimizado de cuarto orden de tipo Chebyshev de líneas acopladas. Asimismo, se muestra la máscara de atenuación. Para validar los resultados se muestra la solución directa (línea con guiones).

Butterworth. El primer filtro Butterworth diseñado posee una frecuencia central en 10 GHz, un ancho de banda relativo del 2,0 %. La atenuación en las bandas de rechazo debe ser superior a 20,0 dB en los

rangos $f \leq 9,70$ GHz y $f \geq 10,30$ GHz. El filtro debe ser de tercer orden para cumplir dichas especificaciones [Hong y Lancaster, 2001]. Las dimensiones del filtro calculadas con las mismas RBFNNs empleadas en los filtros de tipo Chebychev se desglosan en la Tabla 5.10. Con estas dimensiones se obtiene la respuesta mostrada en la Figura 5.45. Al igual que en los filtros de tipo Chebychev, el grado de cumplimiento es muy satisfactorio.

Tabla 5.10: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial.
Filtro de tercer orden de tipo Butterworth de líneas acopladas de la caja de la Figura 5.27.

Parámetro	Sep. inicial en mm	Long. inicial en mm
$Q_e = 49,9975$	0.251	5.6277
$k_{12} = 0,0141$	1.710	5.6277
$k_{23} = 0,0141$	1.710	5.6277

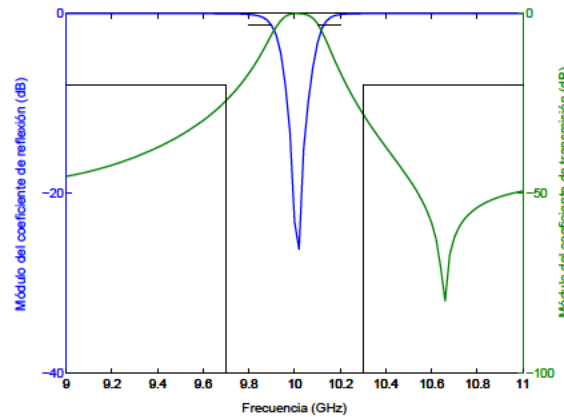


Figura 5.45: Respuesta del filtro inicial de tercer orden de tipo Butterworth de líneas acopladas correspondiente a las dimensiones de la Tabla 5.10. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}).

Gracias a la rapidez del método de análisis neuronal y a la calidad del filtro inicial, es posible aplicar las sencillas reglas mencionadas para obtener un filtro que cumpla completamente las especificaciones. En este filtro, las variables de optimización son la separación entre el puerto y el resonador adyacente (S_1), la separación entre el primer y segundo resonador (S_2), la longitud del primer y tercer resonador (L_B) y la longitud del segundo resonador (L_C). En la Tabla 5.11 se muestran los incrementos aplicados en las variables de optimización. Para optimizar el filtro no fue necesario variar la separación (S_2) y por ello no aparece en la Tabla 5.11. En primer lugar se optó por centrar el filtro en la banda deseada y por ello se actuó sobre las longitudes de los resonadores, ver Figuras 5.46(a) y 5.46(b). Una vez centrada la respuesta del filtro, se procedió a la ampliación del ancho de banda relativo como se aprecia en la Figura 5.46(c) correspondiente a la cuarta prueba. En la sexta iteración el filtro cumple casi todas las especificaciones como se aprecia en la Figura 5.46(d). En esta iteración el filtro pierde un tanto la forma de una respuesta tipo Butterworth. Finalmente, en la iteración séptima se consigue el cumplimiento en toda la máscara de atenuación, ver Figura 5.47. En esta última iteración se aprecia que se acentúa el parecido con una respuesta tipo Chebychev. En

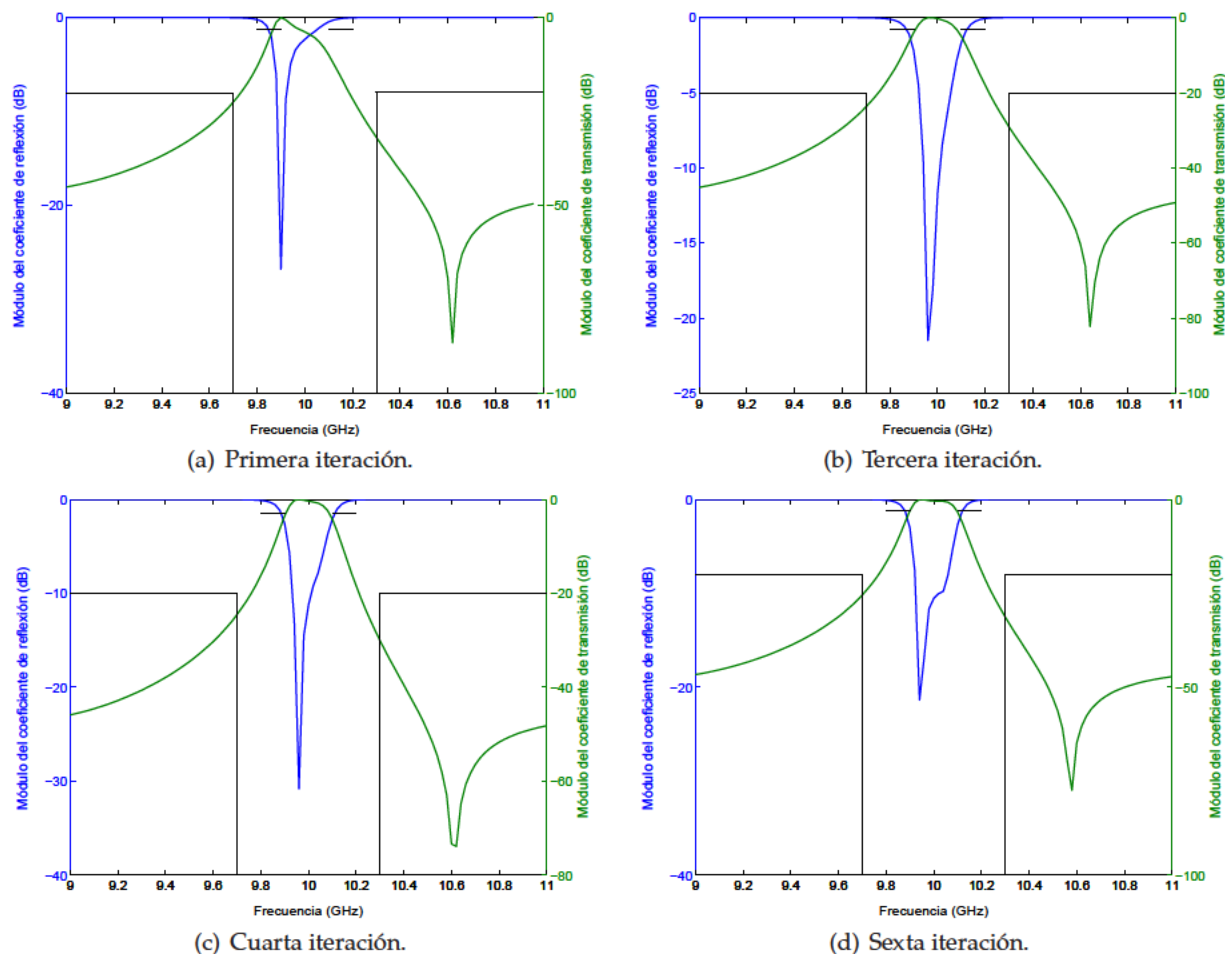


Figura 5.46: Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de tercer orden de Butterworth realizado con líneas acopladas.

los dos filtros de líneas acopladas se han aplicado incrementos positivos pero si fuera necesario se podrían haber empleado incrementos negativos.

Tabla 5.11: Incrementos en mm. aplicados a las separaciones y longitudes iniciales en la optimización el filtro de tercer orden de líneas acopladas de tipo Butterworth de la caja de la Figura 5.27.

Variable	Iteración						
	1	2	3	4	5	6	7
ΔS_1	0.000	0.000	0.000	0.025	0.035	0.055	0.075
$\Delta L_B = \Delta L_C$	0.050	0.025	0.015	0.015	0.015	0.015	0.015

En los resonadores de tipo horquilla el empleo del método de análisis basado en Redes neuronales de Chebychev también reduce el tiempo requerido para generar los conjuntos de entrenamiento y validación. Para entrenar y validar la primera de las RBFNNs de la Figura 5.23 se generaron 104 datos de entrenamiento y 96 datos de validación. El tiempo invertido en esta tarea fue de 2 horas y 5 minutos. El cálculo de los datos de la segunda RBFNN se prolongó durante 3 horas y 9 minutos.

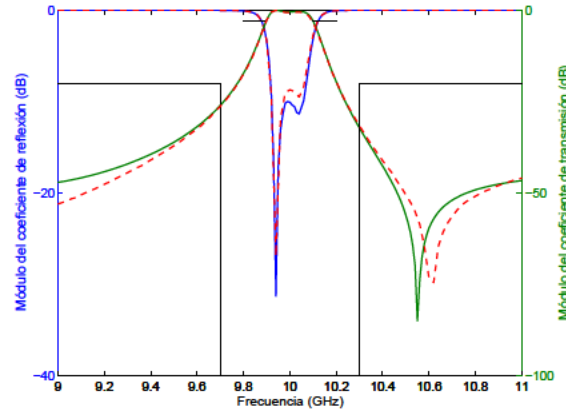


Figura 5.47: Respuesta del filtro optimizado de tercer orden de tipo Butterworth de líneas acopladas. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}). Para validar los resultados se muestra la solución directa (línea con guiones).

En este caso el conjunto de entrenamiento estaba compuesto por 78 valores mientras que se reservaron 72 datos para validar el funcionamiento de la RBFNN. En total, el tiempo transcurrido desde la elección del tipo de resonador hasta la obtención de las RBFNNs conectadas en cascada es de 5 horas y 14 minutos (sin tener en cuenta el tiempo de diseño de las Redes neuronales de Chebychev involucradas en la formación del método neuronal de análisis). Este periodo es mucho menor que el requerido en el caso de emplear el método de análisis basado en RBFNNs (más de 38 horas en total). Además, los nuevos datos de entrenamiento y validación son aptos para calcular las dimensiones de los filtros deseados. Como se aprecia en las Figuras 5.48, 5.49 y 5.50 los datos de validación de las dos RBFNNs conectadas en cascada son similares a los ya obtenidos en la sub-sección anterior. Al igual que en el caso de los resonadores de líneas acopladas se espera una mejora en la calidad del método de diseño gracias a la mayor precisión del método de análisis. En la Tabla 5.12 se muestran las dimensiones del filtro de quinto orden de tipo horquilla calculadas con las RBFNNs re-entrenadas. Todas las separaciones así como la longitud del resonador son similares a las calculadas en la sub-sección previa (ver Tabla 5.6). En la Figura 5.51 se observa la respuesta del filtro inicial obtenida con dichas dimensiones. De nuevo, este filtro constituye un excelente punto inicial para un breve proceso de optimización.

Tabla 5.12: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de quinto orden de tipo horquilla de la caja de la Figura 5.28.

Parámetro	Tap. y sep. iniciales en mm	Long. inicial en mm
$Q_e = 5,755$	6.7480	18.06
$k_{12} = 0,1589$	0.2865	18.06
$k_{23} = 0,1211$	0.5129	18.06
$k_{34} = 0,1211$	0.5129	18.06
$k_{45} = 0,1589$	0.2865	18.06

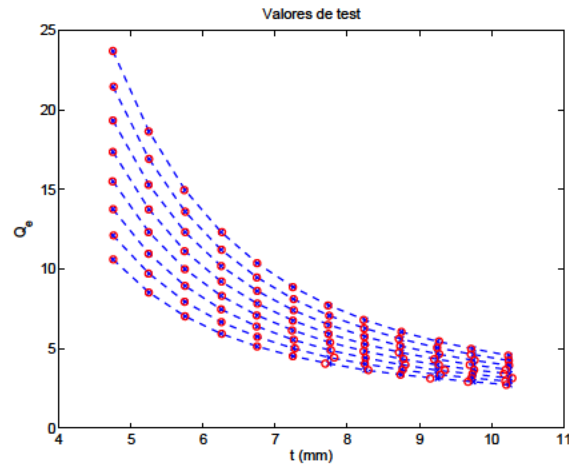


Figura 5.48: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de tipo horquilla. Los factores de calidad han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.19(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

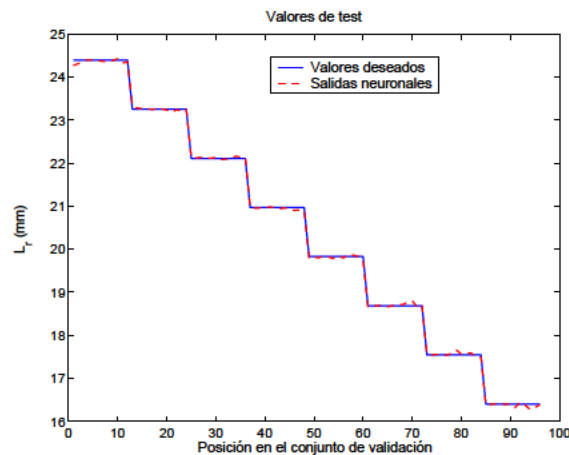


Figura 5.49: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de tipo horquilla. Las frecuencias de resonancia han sido calculadas mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.19(a).

El proceso de optimización directa opera en el filtro de tipo horquilla sobre cuatro variables dibujadas en la Figura 5.28. A saber, la distancia de "tapping" (t), la longitud de los resonadores (L) y las separaciones (S_1) y (S_2). En la Tabla 5.13 se muestran los valores aplicados en las anteriores dimensiones físicas. La separación (S_2) no sufrió ningún cambio durante la fase de optimización directa y por tanto no se muestra. Las primeras dos pruebas sirvieron para centrar un tanto mejor la respuesta del filtro y para ajustar su ancho de banda relativo como se observa en las Figuras 5.52(a)

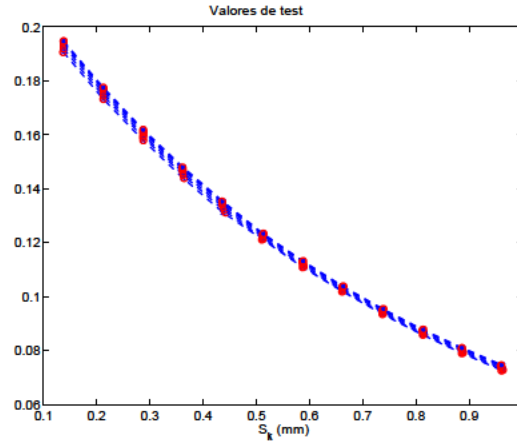


Figura 5.50: Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de tipo horquilla. Los coeficientes de acoplo han sido calculados mediante el métodos neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.19(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.

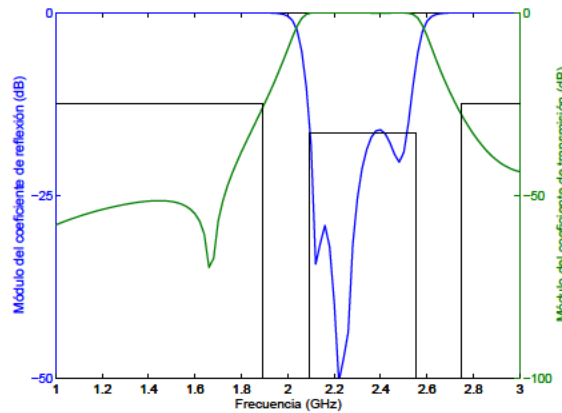


Figura 5.51: Respuesta del filtro inicial de quinto orden formado con resonadores de tipo horquilla correspondiente a las dimensiones de la Tabla 5.6. Asimismo, se muestra la máscara de atenuación.

y 5.52(b). La variación de la separación (S_2) permite, sobre todo a partir de la tercera prueba, ajustar la respuesta para que se cumplan todas las especificaciones, ver Figuras 5.52(c) y 5.52(d). En la sexta iteración el proceso se dio por finalizado al alcanzar la respuesta mostrada en la Figura 5.53. De nuevo, la optimización directa es fácil gracias a que el punto de partida es de gran calidad.

Para probar la adaptabilidad del método desarrollado se procedió a diseñar un filtro de tipo Butterworth con resonadores de tipo horquilla. Se desea que este filtro esté situado en torno a 1,72 GHz, muy cerca del límite inferior del conjunto de entrenamiento. El ancho de banda relativo debe ser del 14,5 % y la atenuación en las bandas de rechazo, $f \leq 1,30$ GHz y $f \geq 2,15$ GHz, debe superar los 30 dBs. Aplicando las fórmulas expuestas en [Hong y Lancaster, 2001] el orden del filtro debe

Tabla 5.13: Valor en mm. aplicado a la longitud de los resonadores (L), la distancia de "tapping" (t) y la separación (S_2) inicial en la optimización el filtro de quinto orden con resonadores de tipo horquilla de la caja de la Figura 5.28.

Variable	Iteración					
	1	2	3	4	5	6
Valor (L)	18.00	18.00	18.00	18.00	18.00	18.00
Valor (t)	5.800	5.800	5.800	5.800	5.800	5.800
Valor (S_2)	0.513	0.500	0.480	0.450	0.460	0.465

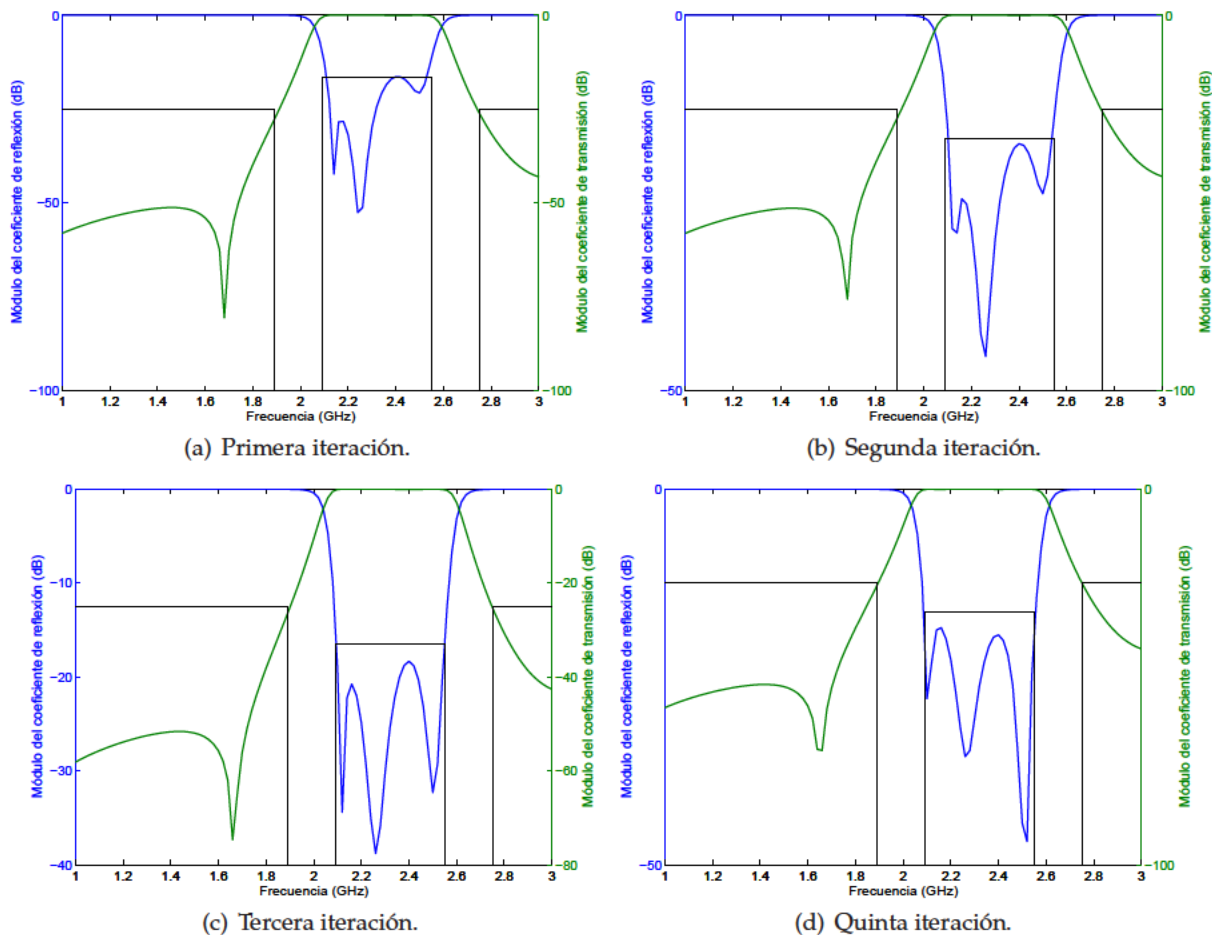


Figura 5.52: Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de quinto orden de Chebychev realizado con resonadores de tipo horquilla.

ser de cuarto orden. Las dimensiones calculadas con las RBFNNs se muestran en la Tabla 5.14. La respuesta del filtro generado con dichas dimensiones se exhibe en la Figura 5.54. Aunque existe un desplazamiento en frecuencia y la respuesta corresponde más a una función de transferencia Chebychev, este filtro constituye un excelente punto de partida para la fase de optimización. En efecto, tan solo se requirieron cuatro pruebas para lograr un completo cumplimiento de las especificaciones, ver Tabla 5.15. Además, solo se actuó sobre la distancia de "tapping" (t) y la longitud común a todos los resonadores (L).

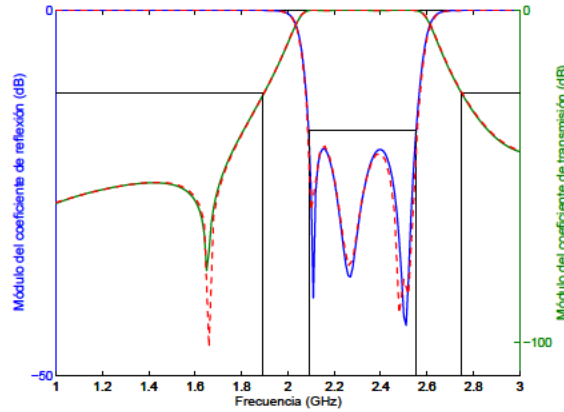


Figura 5.53: Respuesta del filtro optimizado de quinto orden de Chebychev formado por resonadores de tipo horquilla. Asimismo, se muestra la máscara de atenuación. Para validar los resultados se muestra la solución directa (línea con guiones).

Tabla 5.14: Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de cuarto orden de tipo horquilla de la caja de la Figura 5.28.

Parámetro	Tap. y sep. iniciales en mm	Long. inicial en mm
$Q_e = 5,2671$	8.9461	22.987
$k_{12} = 0,1222$	0.5191	22.987
$k_{23} = 0,0786$	0.9097	22.987
$k_{34} = 0,1222$	0.5191	22.987

Con el primer cambio de la longitud (L) ya se consigue que la frecuencia central sea muy cercana a la deseada, ver Figura 5.55(a). La modificación de la distancia (t) posibilita la ampliación del ancho de banda para conseguir la satisfacción de toda la máscara de atenuación, ver Figuras 5.55(b) y 5.55(c). La respuesta final del filtro optimizado se muestra en la Figura 5.56. Esta respuesta no es exactamente de tipo Butterworth pero cumple las especificaciones tanto en la banda de rechazo como en la de paso. En esta figura se aprecia además la exactitud del método de análisis basado en Redes neuronales de Chebychev con respecto a la solución conseguida con la IE directa.

Tabla 5.15: Valor en mm. aplicado a la longitud de los resonadores (L) y la distancia de "tapping" (t) en la optimización el filtro de cuarto orden con resonadores de tipo horquilla de la caja de la Figura 5.28.

Variable	Iteración			
	1	2	3	4
Valor (L)	24.0	24.0	24.0	24.0
Valor (t)	8.00	7.50	7.00	6.75

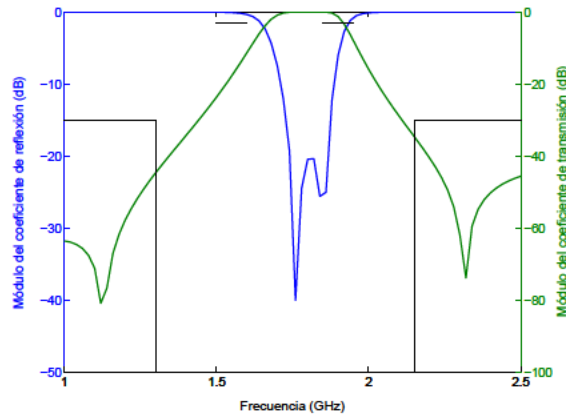


Figura 5.54: Respuesta del filtro inicial de cuarto orden de tipo Butterworth formado con resonadores de tipo horquilla correspondiente a las dimensiones de la Tabla 5.14. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}).

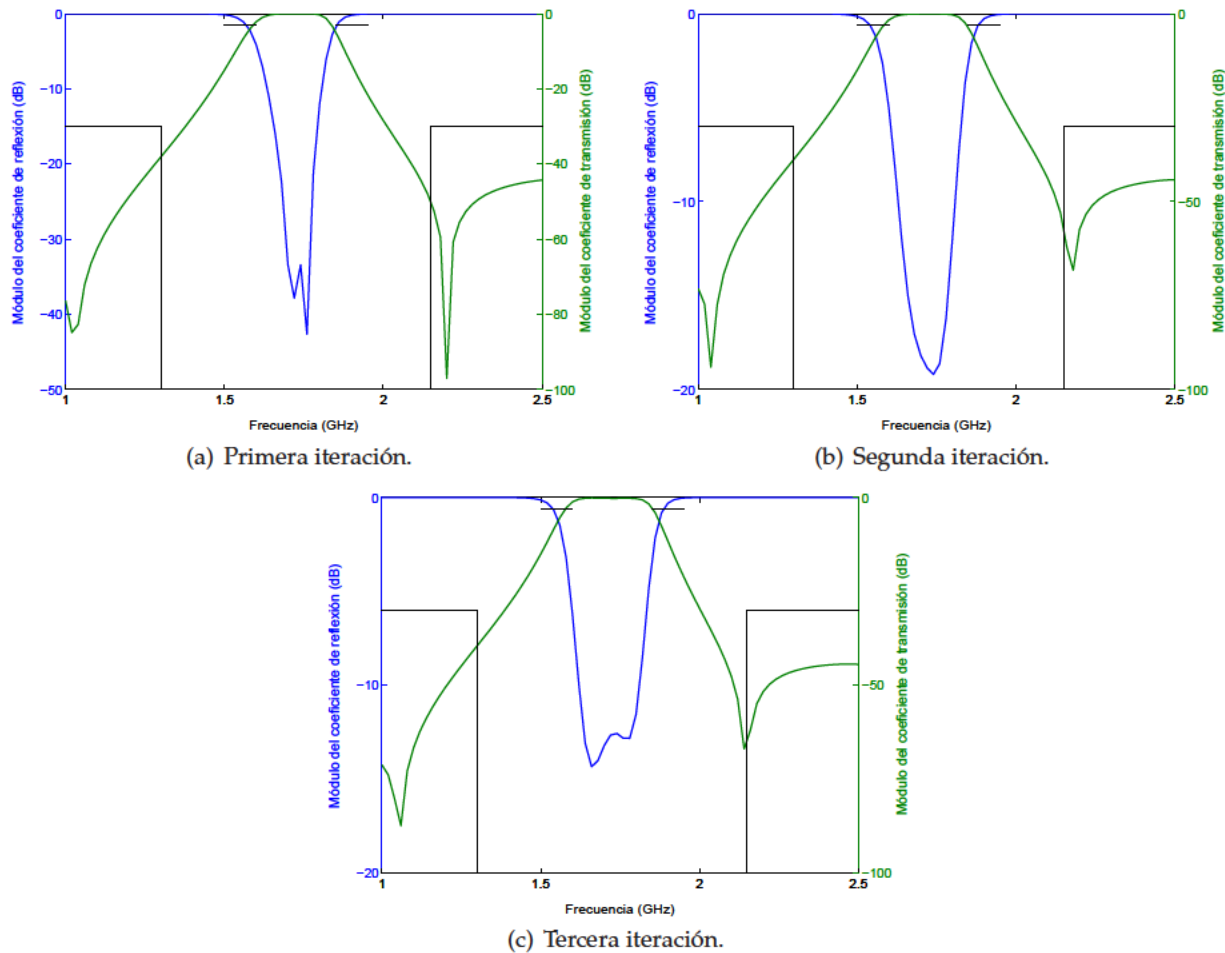


Figura 5.55: Respuesta de tres iteraciones del proceso de optimización directo del filtro de cuarto orden de Butterworth realizado con resonadores de tipo horquilla.

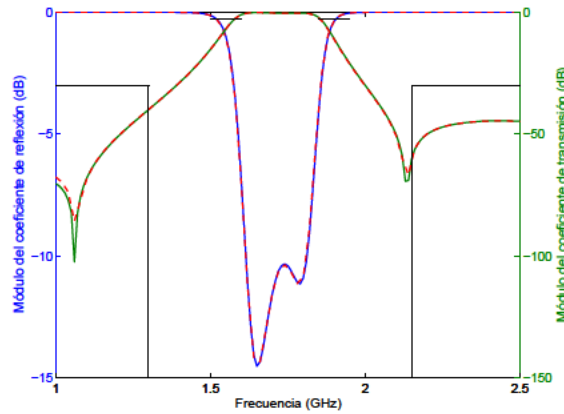


Figura 5.56: Respuesta del filtro optimizado de cuarto orden de tipo Butterworth formado por resonadores de tipo horquilla. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}). Para validar los resultados se muestra la solución directa (línea con guiones).

5.4. Conclusiones

Las dos técnicas desarrolladas han obtenido resultados satisfactorios en el diseño de filtros impresos en una estructura apantallada multicapa. La primera técnica utiliza un Algoritmo genético para optimizar un filtro inicial que no cumple las especificaciones fijadas. Este método presenta dos novedades fundamentales. En primer lugar, utiliza el método de análisis neuronal descrito en [Garcia et al., 2006b]. Gracias a la velocidad de las RBFNNs, una vez entrenadas, la técnica de los Algoritmos genéticos se puede aplicar de forma práctica. En segundo lugar, se ha desarrollado una nueva función de rendimiento que permite cuantificar de forma apropiada el cumplimiento de las especificaciones por parte de cada individuo de la población. Esta es una tarea esencial para que el funcionamiento del Algoritmo genético sea correcto. Para lograr este objetivo, la función de rendimiento está compuesta por funciones logísticas. La función de rendimiento aporta diferentes beneficios al Algoritmo genético. Por un lado, asegura que el extremo encontrado constituya una solución válida del problema, es decir, que efectivamente satisfaga las especificaciones. Por otro lado, si los parámetros libres de las funciones logísticas son seleccionados de forma adecuada, se impide la aparición del problema de la convergencia prematura. Para perfeccionar la función de rendimiento desarrollada, se ha implantado un método automático de cálculo de los parámetros libres. De esta forma, se evita la intervención del diseñador que ya no tendrá que fijar de forma heurística, en base a su experiencia, dichos parámetros libres. Además, se asegura que la función de rendimiento no producirá convergencia prematura y que representará adecuadamente la calidad del filtro. La mencionada función del AG equivale a las funciones de error utilizadas en las técnicas de optimización por gradiente. En las funciones de error, para cada frecuencia, se calcula la diferencia entre la respuesta del filtro y las especificaciones. Las operaciones efectuadas sobre la mencionada diferencia forman la función de error [Chen, 1988]. Como trabajo futuro se plantea la utilización de las funciones de error provenientes del campo de la optimización de circuitos mediante técnicas de gradiente en la técnica de los AG.

Como se ha visto en los ejemplos, la convergencia de las sucesivas generaciones hasta una solución es firme. Sin embargo, conforme el problema de optimización se torna más complicado la op-

timización es más lenta. Además, en el filtro de cuarto orden las soluciones alcanzadas, a pesar de presentar una respuesta muy cercana a la respuesta óptima, no cumplen totalmente la máscara de atenuación propuesta. Una posible solución radica en tomar la mejor solución aportada en la última generación del AG y emplearla como punto de partida de un proceso de optimización por gradiente. Al satisfacer casi completamente las especificaciones es muy probable que la optimización del mejor filtro sea sencilla y rápida. Como han mostrado los ejemplos, el Algoritmo genético invierte mucho tiempo en la búsqueda de la solución. Es cierto que en la Sección 5.2 se ha utilizado el método neuronal basado en RBFNNs, zonas cuadradas y multiplicación por (ρ) que es más lento que el fundamentado en Redes neuronales de Chebychev. Sin embargo, incluso la utilización del método de análisis neuronal más rápido puede conducir a periodos de operación de horas o decenas de horas dependiendo de la complejidad del problema. Esta inversión es rentable si el filtro inicial disponible se encuentra muy alejado del cumplimiento de alguna de las especificaciones. Si el filtro no se encuentra alejado de la respuesta buscada, entonces no es aconsejable utilizar el Algoritmo genético. Una técnica basada en el uso del gradiente presentará mejores resultados en un tiempo menor cuando se disponga de un buen punto de partida. Por supuesto, existen soluciones intermedias fruto de la combinación de ambas técnicas. Si no se está seguro sobre el éxito de las técnicas basadas en el gradiente, se puede iniciar una fase de optimización con el AG para dar paso, tras pocas generaciones, a la fase de optimización por gradiente. Esta última técnica se aplicaría a los mejores filtros de la población, no solo al mejor, ya que alguno de ellos puede conducir al extremo global. La combinación de forma ordenada de ambas técnicas suele denominarse "escalada de las montañas" [Goldberg, 1989].

Si se deben diseñar un número significativo de filtros impresos en una determinada estructura multicapa, entonces resulta muy rentable emplear la segunda de las técnicas desarrolladas. Este método utiliza RBFNNs para aprender las relaciones entre los parámetros eléctricos y las dimensiones de los resonadores que los sintetizan. Este es un método previamente empleado en otras tecnologías como la de guíaonda. Las principales novedades de nuestra técnica radican en la aplicación del método en estructuras apantalladas multicapa y en el uso de RBFNNs, mucho más eficientes que los tradicionales MLPs. Para generar los conjuntos de entrenamiento y validación requeridos por las redes neuronales es necesario invertir un tiempo inicial. Conforme se diseñan más filtros en la estructura y en el ancho de banda de trabajo, más rentable deviene el método desarrollado. En este caso, se aprecia la gran ganancia temporal lograda por el método de análisis basado en Redes neuronales de Chebychev. Mientras que el método basado en RBFNNs de [Garcia et al., 2006b] tarda unas decenas de horas en generar los conjuntos mencionados, el método neuronal de Chebychev tan solo tarda unas pocas horas. Es cierto que en la rapidez de este segundo método influye la utilización de la técnica de extracción de dos capas para cancelar la singularidad, pero de todas formas buena parte de la ganancia corresponde a la habilidad del método de entrenamiento propuesto en las Redes neuronales de Chebychev para seleccionar sólo los mejores regresores.

En cualquier caso, el periodo de generación de los conjuntos de datos y el tiempo de entrenamiento son una fase inicial del problema que no se debe incluir en el tiempo de diseño del filtro. Las dimensiones son obtenidas en las RBFNNs de forma instantánea. Como se ha visto en los ejemplos propuestos, es necesario aplicar un rápido proceso de optimización al filtro obtenido con las mencionadas dimensiones, ya que éste no cumple todas las especificaciones. Este hecho no parece deberse a errores significativos en la aproximación de las curvas por medio de las RBFNNs. El error

NMSE máximo fijado es muy pequeño y como se aprecia en las figuras de la Sub-sección 5.3.3 y de la Sub-sección 5.3.5 las RBFNNs arrojan valores muy cercanos a los valores de validación. Asimismo, los parámetros eléctricos han sido calculados de un modo apropiado. Las estructuras de la Figura 5.17 para los resonadores de líneas acopladas y de la Figura 5.19 para los resonadores tipo horquilla permiten la evaluación de los factores de calidad externos, los coeficientes de acoplo y la frecuencia de resonancia. La principal fuente de error proviene de los acoplos introducidos entre resonadores no adyacentes cuando todos los resonadores son incluidos en el filtro. Estos acoplos alteran la respuesta del filtro de modo que ésta se aleja de la ideal. Este es un defecto que también ocurre en aquellos filtros diseñados a partir de tablas o curvas.

A pesar de los defectos del método, el cumplimiento de las especificaciones es muy elevado. Por ello, el proceso de optimización es muy sencillo. Ocupa tan solo pocas iteraciones de un algoritmo basado en el gradiente. Incluso se puede ajustar de forma directa mediante la aplicación de ciertas reglas. La existencia de estas reglas y el éxito conseguido con la optimización directa, hacen suponer que es factible la aplicación de técnicas de ajuste basadas en lógica difusa. Esta constituye una línea futura de gran interés, ya que la lógica difusa no solo aplicaría las reglas ya conocidas, sino que permitiría encontrar reglas más complejas a partir de los datos que relacionan los cambios en la respuesta con los cambios en las dimensiones.

Capítulo 6

Conclusiones y líneas futuras

En la presente tesis se ha desarrollado una herramienta de análisis electromagnético de circuitos multicapa apantallados basada en la aproximación de las funciones de Green de los potenciales del medio mediante redes neuronales. Esta aproximación es posible gracias a que las redes neuronales son aproximadores multidimensionales que capturan las relaciones no lineales entre un espacio de entrada y un espacio de salida. Después del entrenamiento de las redes neuronales para una estructura apantallada en particular, éstas sustituyen a las funciones de Green evaluadas numéricamente en la resolución de la Ecuación integral (IE) mediante el Método de los momentos (MoM). En particular, se ha empleado una formulación espacial de las funciones de Green y de la IE. El uso de las redes neuronales ha permitido la aceleración del análisis circuital en gran medida. Gracias a la veloz herramienta de análisis se han desarrollado dos nuevas técnicas de diseño de filtros multicapa apantallados. El primer método emplea Algoritmos genéticos para optimizar el rendimiento de un filtro inicial que no cumple las especificaciones de diseño. El segundo método utiliza redes neuronales para aproximar las relaciones existentes entre ciertos parámetros eléctricos del filtro deseado y las dimensiones físicas que los sintetizan. En ambos casos se han diseñado con éxito diversos filtros con diferentes características.

La aproximación de las funciones de Green de los potenciales relevantes del medio es una tarea muy complicada. La presencia de la singularidad en el punto donde se encuentra la fuente produce un fuerte crecimiento de las funciones de Green en torno a ella. Para conseguir una aproximación, siquiera imperfecta, es necesario extraer la singularidad a la fuente. Cuanto mejor es la eliminación de dicha singularidad, menos abrupto es el comportamiento de las funciones deseadas y por tanto mejor es la aproximación. Cuando las redes neuronales son utilizadas en la resolución de la IE, se recupera la singularidad para recomponer de forma completa las funciones de Green. En esta tesis se han empleado dos procedimientos diferentes para extraer la singularidad. El primero consiste en multiplicar las funciones de Green por la distancia (ρ) entre punto fuente y punto observación. De esta forma, para recuperar las funciones de Green originales tan solo hay que dividir por la distancia (ρ) los valores de salida de la red neuronal. Esta técnica fue empleada para poner a prueba el rendimiento de tres tipos de redes neuronales en la aproximación de las funciones de Green. En concreto se utilizaron las Redes neuronales de funciones de base radial (RBFNN), el Perceptrón multicapa (MLP) y las Máquinas de soporte vectorial (SVM). Las RBFNNs alcanzaron la mayor precisión;

además también consiguieron la mejor relación entre el tiempo de entrenamiento, el nivel de error en la aproximación y el número de neuronas. El tamaño neuronal de la red neuronal es fundamental para que el método neuronal de análisis sea rentable frente a las técnicas que emplean las funciones de Green exactas. Si el tamaño de las redes neuronales es reducido, entonces el análisis circuital será muy rápido. Respecto al tiempo de entrenamiento, aunque constituye un periodo inicial, cuanto menor sea más provechoso será el método neuronal. Por todo ello, se seleccionó la RBFNN frente a las otras dos redes neuronales (MLP y SVM). En cada red neuronal, existen distintos métodos de entrenamiento, es decir, diferentes formas de calcular los parámetros neuronales. En la RBFNN, se han comprobado dos métodos no supervisados, como la selección aleatoria de centros y el algoritmo de K-medias, y un método supervisado como el algoritmo OLS. Debido a que, incluso una vez extraída la singularidad, la aproximación de las funciones de Green es un problema complejo, el algoritmo OLS alcanza los mejores resultados.

De la experiencia obtenida en la aproximación de las funciones de Green podemos concluir que si la función objetivo es de difícil aproximación, la mayor precisión del método OLS compensa con creces el hecho de que su tiempo de entrenamiento sea mucho más prolongado que el del resto de técnicas de entrenamiento tratadas. De nada sirve que el periodo de entrenamiento sea reducido, si la aproximación de las funciones de Green no es fiable o si el tamaño neuronal debe crecer desmesuradamente para alcanzar la precisión deseada. El algoritmo OLS consume muchos recursos de computación, tanto en términos de tiempo como de memoria, pero permite la generación de RBFNNs más compactas y más precisas. Si las funciones que se desean representar son de fácil aproximación y requieren de conjuntos de entrenamiento grandes, entonces puede ser conveniente el uso de técnicas sencillas como la de K-medias. Dependiendo de la aplicación de la RBFNN, hay que seleccionar una u otra técnica de entrenamiento. Si el problema cuenta con conjuntos de entrenamiento pequeños, tanto los algoritmos supervisados (OLS) como los no supervisados pueden ser empleados con resultados satisfactorios. Así ocurre en la captura de las relaciones entre los parámetros eléctricos y dimensiones físicas en la segunda técnica de diseño.

Las RBFNNs deben calcular las salidas apropiadas para cualquier pareja de puntos fuente y observación. El espacio de entrada cuenta así con cuatro dimensiones correspondientes a las coordenadas transversales del punto fuente (x', y') y las coordenadas transversales del punto observación (x, y). Además, también se ha optado por añadir como dimensión de entrada la frecuencia de análisis, de modo que las redes neuronales aproximan las funciones de Green para una estructura dada a lo largo de un ancho de banda determinado. El espacio de entrada cuenta así con cinco dimensiones en total. La coordenada longitudinal tanto del punto fuente (z') como del punto observación (z) no han sido incluidas ya que solo se ha trabajado con estructuras planares en las que no existen fuentes longitudinales, es decir, sin metalizaciones verticales. Por ello y debido a la elección de los potenciales de Sommerfeld, las funciones que deben ser aproximadas son las partes reales e imaginarias de G_A^{xx} , G_A^{yy} y G_V . Por la ausencia de fuentes en el eje z o eje longitudinal, no es necesario calcular las componentes G_A^{zx} , G_A^{zy} y G_A^{zz} y por tanto no forman parte del espacio de salida. Si existiesen metalizaciones verticales, entonces las salidas de las redes neuronales serían las partes reales e imaginarias de G_A^{xx} , G_A^{yy} , G_A^{zx} , G_A^{zy} , G_A^{zz} y G_V mientras que el espacio de entrada debería incorporar las componentes longitudinales del punto fuente y observación (z, z'). El problema de aproximación se convertiría en más difícil, principalmente por el aumento de la dimensión total del espacio de entrada, pero la

metodología sería la misma que la seguida en las estructuras tratadas en el trabajo de esta tesis.

En la tesis se ha trabajado con estructuras planares encapsuladas compuestas por una capa de dieléctrico y una capa de aire (microtira encapsulado). El actual método neuronal puede ser aplicado a estructuras más complejas, compuestas por varias capas de dieléctrico. Una de las líneas de trabajo futuras es sin duda la ampliación del método desarrollado a este tipo de estructuras compuestas por más capas de dieléctrico. Para ello es necesario aproximar las funciones de Green no solo cuando el punto fuente y punto de observación se encuentren en la misma capa sino también cuando se ubiquen en diferentes capas. Si, por ejemplo, la estructura multicapa estuviera formada por dos capas de dieléctrico en el que se van a imprimir metalizaciones, habría que aplicar el método neuronal descrito tres veces. A saber, en la situación en la que el punto fuente y observación se encontrasen en la primera capa de dieléctrico, cuando ambos puntos se encontrasen en la segunda capa, y finalmente cuando el punto fuente se ubicase en la primera capa y el punto observación en la segunda capa. La situación en la que el punto fuente se localizase en la segunda capa y el punto observación en la primera capa es recíproca del último caso. El tiempo invertido en el diseño de las redes neuronales aumentará considerablemente, pero tan solo se trata de una repetición de la misma metodología. Al incrementarse este periodo inicial de diseño, el método neuronal perdería interés únicamente si en una estructura dada se analizaran pocos circuitos impresos, o se realizasen pocos análisis de un mismo circuito.

La aproximación de las funciones de Green requiere de conjuntos de entrenamiento muy extensos ya que las redes neuronales deben expresar las relaciones entre el espacio de salida y el espacio de entrada de forma completa. A pesar de las propiedades de simetría de las funciones de Green en las cavidades rectangulares, que producen una limitación en los valores que pueden tomar las coordenadas del punto fuente, el espacio de entrada es muy extenso. Además, el espacio de entrada debe ser muestreado de forma completa y densa para que las redes neuronales aproximen de forma precisa las funciones de Green, especialmente cerca de la fuente y cerca de las paredes metálicas, donde se deben cumplir las condiciones de contorno de los campos y potenciales. La solución propuesta en esta tesis ha consistido en dividir el espacio de entrada en zonas, de modo que cada RBFNN está especializada en la aproximación de las funciones de Green en una única región del espacio de entrada. Como se ha demostrado en la tesis esta técnica ha permitido la correcta aproximación en todo el espacio de entrada. Como consecuencia, múltiples filtros han sido analizados con éxito en cada una de las estructuras utilizadas. La forma de las zonas espaciales no es arbitraria. Así, las zonas fuente se adaptan a la forma de la cavidad y por ello tienen forma de "L". De este modo las condiciones de contorno cerca de las paredes son expresadas de forma más apropiada. Las zonas de puntos observación son más pequeñas cerca de la fuente, por lo que la densidad de puntos es mayor. Así, las variaciones en el entorno de la fuente se representan mejor. La coordenada frecuencia es asimismo dividida en diversas zonas. Dicha partición posibilita la aplicación del método neuronal en estructuras encapsuladas en las que aparezcan frecuencias de resonancia.

Para facilitar la utilización del método neuronal, el algoritmo de división es automático. La generación de las zonas observación comienza con la zona más cercana a la fuente, mientras que la generación de las zonas fuente empieza con la zona más alejada de las paredes. En todas las cajas metálicas tratadas, la aproximación de las funciones de Green es más difícil conforme las zonas fuente se ubican más cerca de las paredes. Debido a que la determinación de las zonas observación

y de las zonas fuente está relacionada, las primeras son fijadas con la zona fuente más alejada de las paredes. Esta solución ha funcionado en todas las cavidades, pero un procedimiento más adecuado y que permitiría la aproximación de las funciones de Green de forma más exigente consistiría en comenzar la división del espacio de entrada desde la región fuente más próxima a las paredes metálicas. No obstante, la división del espacio de entrada ha constituido un gran acierto en el problema de la aproximación de las funciones de Green. Principalmente, la división proporciona conjuntos de entrenamiento reducidos. Así, el algoritmo supervisado OLS puede ser empleado sin que aparezcan problemas de saturación de memoria al almacenar la matriz de regresores. Además, los conjuntos de entrenamiento pequeños implican tiempos de entrenamiento menores y redes neuronales más compactas, que llevan aparejadas tiempos de cálculo reducidos. Por tanto, todo el método neuronal es más atractivo gracias al procedimiento de división aplicado. El algoritmo de división del espacio de entrada fue completado con un algoritmo adaptativo de búsqueda de la varianza óptima en las funciones gaussianas de las neuronas de las RBFNNs. Como se ha comprobado en los ejemplos mostrados en la tesis, la elección del valor de varianza es fundamental para que la aproximación sea precisa. Generalmente, las zonas observación situadas cerca de la fuente requieren valores de varianza menores ya que las funciones objetivo crecen rápidamente. Lejos de la fuente, la varianza puede tomar valores mayores debido a que las variaciones son mucho más lentas. El algoritmo de búsqueda de varianza es automático; así se libera al operador humano de la tarea de fijación de las varianzas en cada zona observación.

La técnica de extracción de la singularidad basada en la multiplicación por la distancia (ρ) combinada con la división del espacio de entrada y el algoritmo de búsqueda de la varianza óptima posibilita el análisis de filtros que presentan acoplos críticos. Sin embargo, aunque el espacio de entrada sea dividido en multitud de zonas y se encuentre una varianza apta para cada zona observación, existe un margen de mejora en la precisión del método neuronal que emplea el método de extracción de la singularidad comentado. Este hecho es debido a que la recuperación de la función de Green original produce una amplificación del error cometido por las RBFNNs. El aumento de la precisión es deseable en las cercanías de la fuente debido el Método de los momentos evalúa las funciones de Green muchas más veces en esta región que en aquellas regiones de puntos observación alejadas de la fuente. La integración de las interacciones entre funciones base cercanas o de una función de base con ella misma conlleva el uso de un mayor número de puntos de cubatura, debido justamente a la presencia de la singularidad de la fuente.

Para incrementar drásticamente la precisión se ha aplicado un método de extracción de la singularidad que emplea las peculiaridades del método numérico aplicado para calcular las funciones de Green. La evaluación numérica requiere de la división del espacio de puntos observación, para un punto fuente determinado, en dos regiones: la región de máxima singularidad (MS) y la región de débil singularidad (DS). En la región MS se recurre a una suma de imágenes espaciales. En la región DS la suma modal es acelerada mediante la técnica de suma por partes. La extracción, del sumatorio total de imágenes, de las dos capas de conjuntos de imágenes espaciales más cercanas a la fuente, permite la obtención de funciones sin ningún comportamiento singular y de lenta variación. Como se ha comprobado en los ejemplos y tests realizados en la tesis, la nueva técnica de eliminación de la singularidad permite que la zona más cercana a la fuente pase a convertirse en la zona observación con mayor precisión en la aproximación. Una vez es recobrada la singularidad, las difer-

encias entre la precisión alcanzada con la primera técnica y la segunda son todavía más acentuadas ya que la recuperación no produce ninguna amplificación del error. Para recuperar las funciones de Green originales en la región MS durante la ejecución del MoM es necesario sumar a las salidas de las RBFNNs las dos primeras capas de imágenes espaciales (capas de influencia). Esta operación es muy sencilla y no retarda significativamente el análisis circuital ya que es un sumatorio analítico y finito (de unos pocos términos). Los análisis realizados en diferentes cavidades respaldan las anteriores afirmaciones. Para incrementar en mayor medida la alta precisión alcanzada con la nueva técnica, se han utilizado zonas observación circulares en las cercanías de la fuente (región MS). Estas zonas representan de forma más adecuada la simetría esférica impuesta por la fuente. Los resultados obtenidos permiten asegurar que, a pesar de que el uso de estas zonas es positivo, la nueva técnica de eliminación de singularidad es la principal responsable de la mejora en la precisión del análisis circuital. Además de las nuevas zonas circulares, se desarrolló una versión del algoritmo OLS que reduce en gran medida el tiempo de entrenamiento, de modo que convierte en más atractivo al método neuronal. La nueva versión es especialmente adecuada para los tamaños de entrenamiento y número máximo de neuronas utilizados en el método programado para la aproximación de las funciones de Green.

El algoritmo de división del espacio de entrada ha sido aplicado con estas nuevas técnicas del mismo modo que en la técnica de multiplicación por (ρ) . La primera de las zonas observación calculadas es aquella ubicada más cerca de la fuente. Este procedimiento posee más sentido si la aproximación de mayor dificultad se produce cerca de la fuente. Con la nueva técnica de extracción de la singularidad sería más apropiado aplicar un algoritmo de división de las zonas observación de la región MS que comience por el límite entre las regiones MS y DS, y continúe hasta alcanzar las cercanías de la fuente. Aunque los resultados conseguidos con el actual algoritmo de partición son satisfactorios, probablemente el nuevo método de partición obtendría un conjunto de zonas observación más apropiadas. Este hecho redundaría a su vez en una división más eficiente del espacio de puntos fuente, ya que dicho espacio es dividido para cada una de las zonas observación.

El método neuronal que emplea RBFNNs es una herramienta útil de análisis de circuitos multicapa apantallados. El método requiere la inversión de un tiempo inicial en la generación de las zonas observación, fuente y frecuencia, y en el entrenamiento posterior de las RBFNNs especializadas en cada una de las porciones en que es dividido el espacio de entrada. La ganancia temporal conseguida con el método neuronal respecto al método tradicional que hace uso de las funciones de Green exactas (evaluadas numéricamente) es tal, que tan solo hacen falta unos pocos análisis circuitales para alcanzar la rentabilidad deseada. Además, hay que tener en cuenta que las redes neuronales solo deben ser entrenadas una vez para una estructura y ancho de banda determinados. Después del entrenamiento, todas las RBFNNs quedan fijadas, de modo que múltiples circuitos impresos en cualquier zona de la superficie transversal pueden ser analizados con rapidez. El método neuronal que utiliza la técnica de eliminación de la singularidad que extrae dos capas de influencia no solo es el más preciso, sino que también es más veloz que el primer método neuronal. De este modo, se consigue una aceleración del análisis circuital cifrado en más de dos órdenes de magnitud. Aunque el periodo de generación y entrenamiento sea muy elevado (en torno a 50 horas), tan solo un análisis con la solución directa puede alcanzar un tiempo similar, mientras que el análisis con el método neuronal tarda solamente unos minutos.

En la presente tesis no solo se han empleado redes neuronales y métodos de entrenamiento conocidos (como el método OLS en las RBFNNs) en la aproximación de las funciones de Green. La necesidad de una mayor aceleración del análisis circuital ha dado como fruto la aplicación del algoritmo OLS a las Redes neuronales de Chebychev. Hasta donde el autor conoce por la literatura consultada, la utilización de este algoritmo en este tipo de redes neuronales es novedosa. Además, el nuevo muestreo aplicado en el espacio de entrada posibilita un rendimiento óptimo del algoritmo. Las Redes neuronales de Chebychev poseen una estructura similar a la de las RBFNNs. La diferencia fundamental estriba en el empleo de polinomios de Chebychev como funciones neuronales en el caso de un espacio de entrada unidimensional y productorios de polinomios de Chebychev en el caso multidimensional. Un conjunto de polinomios de Chebychev poseen la propiedad de ortogonalidad discreta entre todos ellos en las raíces del polinomio de orden superior. De esta forma, si el espacio de entrada de la red neuronal de Chebychev es muestreado justamente en las raíces de los polinomios de orden superior de cada coordenada, la evaluación del conjunto de entrenamiento en las neuronas producirá un conjunto de vectores regresores inicialmente ortogonales. Como se ha mostrado en la tesis, si los regresores no son ortogonales desde el inicio de la aplicación del algoritmo OLS, éste presenta un rendimiento sub-óptimo. Debido a este hecho, en el caso de las RBFNNs siempre se seleccionarán más neuronas de las necesarias para superar un determinado error de aproximación. En cambio, gracias al apropiado muestreo del espacio de entrada y a la adecuada construcción de las funciones neuronales, en las redes de Chebychev la contribución de un regresor a la expresión de las funciones de salida es calculado de forma inequívoca. Esto es debido a que dicha contribución es independiente de la del resto de regresores.

En la tesis se ha demostrado que las funciones neuronales en el caso unidimensional y multidimensional conducen a regresores ortogonales siempre y cuando el conjunto de entrenamiento sea muestreado en los puntos apropiados. No solo se ha demostrado esta característica de las Redes neuronales de Chebychev; también se ha empleado el teorema de Stone-Weierstrass para demostrar que la red neuronal de Chebychev es un aproximador universal. Por lo tanto, la red neuronal de Chebychev puede aproximar cualquier función continua en el rango de existencia de los polinomios de Chebychev con el grado de precisión deseado. Además, gracias a la ortogonalidad de los regresores, el número de neuronas que alcanzan esa precisión será mínimo (dentro de todas las posibles combinaciones de regresores que tratan de minimizar el término de error utilizado en el algoritmo OLS). La capacidad de aproximación de las redes de Chebychev entrenadas con el algoritmo OLS ha sido puesta a prueba en numerosos tests, consiguiendo excelentes resultados. Gracias a la forma de los polinomios de Chebychev y de sus productos, las nuevas redes neuronales pueden aproximar funciones tanto de lenta variación como de rápida variación. Además, las pruebas indicadas muestran que las Redes neuronales de Chebychev son mucho más resistentes a la aparición del problema de sobre-entrenamiento que las RBFNNs. Cuando las RBFNNs deben aproximar funciones de rápido crecimiento, es adecuado utilizar valores pequeños de varianza, de modo que existe el peligro de convertir las funciones gaussianas en deltas de Dirac situadas en los puntos de entrenamiento. Esta elección conducirá a fuertes aumentos del error de validación frente al error de entrenamiento. En cambio, los polinomios de Chebychev poseen valores distintos de cero en todo el rango de trabajo y su valor está comprendido entre -1 y 1 . Estas características permiten que el sobre-entrenamiento sea poco usual, y cuando exista, que sea de poca envergadura.

En la aproximación de las funciones de Green con Redes neuronales de Chebychev, debido a las particularidades del muestreo en el conjunto de entrenamiento, no se pueden utilizar zonas circulares de observación. Todas las zonas deben ser puramente rectangulares. De este modo, cada zona observación rectangular intermedia utilizada en las RBFNNs debe ser dividida en cuatro sub-zonas. Asimismo, cada zona fuente en forma de "L" debe ser dividida en dos sub-zonas. Esta limitación conlleva una división más intensa del espacio de entrada. Este hecho podría conducir a una merma del atractivo del método neuronal basado en redes de Chebychev. Sin embargo, el algoritmo OLS es mucho más rápido en las redes de Chebychev que en las RBFNNs gracias a las simplificaciones que implican la nulidad de los productos cruzados entre regresores ortogonales. Por ello, la inicial desventaja queda más que compensada. Como ventaja adicional, los polinomios de Chebychev no poseen parámetros que deban ser fijados antes del inicio del algoritmo OLS, como ocurría con la varianza de las gaussianas en las RBFNNs. Así, no es necesario aplicar ningún algoritmo de ajuste de parámetros, lo que permite una cierta simplificación en la división del espacio de entrada, que también contribuye a compensar la inicial desventaja. Además, en las nuevas redes neuronales no es necesario almacenar ninguna matriz de regresores o de producto de regresores como ocurría en las RBFNNs; es decir, desaparecen los problemas de memoria y por tanto la limitación en el tamaño del conjunto de entrenamiento.

Una característica particular del muestreo en las Redes neuronales de Chebychev radica en que los puntos de entrenamiento nunca alcanzan la frontera de la zona a la que pertenecen. Por ello, en dicha frontera, la red neuronal de Chebychev está extrapolando, en lugar de interpolando, a partir de las relaciones no lineales aprendidas. Para poner a prueba de forma exigente la capacidad de aproximación de las Redes neuronales de Chebychev, en todos los conjuntos de validación existen puntos de validación a lo largo de los límites entre zonas. Por esta razón y con el objetivo de probar de modo exigente la capacidad de generalización de las redes de Chebychev, los conjuntos de validación son mucho mayores que los de entrenamiento. Una clara mejora en las Redes neuronales de Chebychev consistiría en que los puntos de muestreo donde los regresores son ortogonales pudieran ser fijados de forma libre, y no fijados por las raíces de los polinomios de orden superior. De esta forma, los puntos de entrenamiento se podrían ubicar en la frontera de las zonas, de modo que el conjunto de entrenamiento encerraría por completo al conjunto de validación y ante cualquier nuevo dato se produciría una interpolación. En los trabajos de Chebychev se explican métodos para generar un conjunto de polinomios ortogonales en una serie de puntos arbitrarios. No obstante, los polinomios generados de esta forma se calculan de forma recursiva, por lo que la generación de los valores de las neuronas sería muy lenta. Es necesario hallar fórmulas simples, como las trigonométricas empleadas en los polinomios de Chebychev de primera especie, que permitan calcular las salidas neuronales de forma rápida.

Las Redes neuronales de Chebychev aproximan las funciones de Green de forma correcta tanto si se utiliza la técnica de eliminación de la singularidad basada en la multiplicación por la distancia (ρ) como si se emplea la extracción de dos capas de influencia. Al igual que en las RBFNNs, ésta última técnica es la más eficiente en la extracción de la singularidad. Por ello, el método neuronal fundamentado en las Redes neuronales de Chebychev usa la segunda técnica para analizar los diferentes filtros. La precisión alcanzada es similar a la conseguida con las RBFNNs, con la ventaja fundamental de una mayor velocidad en el análisis. Esta es una consecuencia directa del rendimien-

to óptimo del algoritmo OLS en las redes de Chebychev. Las redes neuronales son más compactas, y están formadas por menos neuronas. Por ello, el método neuronal de Chebychev es hasta diez veces más rápido que el método neuronal basado en RBFNNs. De esta forma, el nuevo método es competitivo frente a herramientas de análisis como *ADS*® (versión del 2006) u otras formulaciones de la IE fundamentadas en el dominio espectral. Mientras que el método basado en RBFNNs es relativamente poco rentable frente a *ADS*®, el método neuronal de Chebychev, incluso a pesar del tiempo de división y entrenamiento invertido inicialmente, es competitivo frente a la herramienta comercial. Respecto a la formulación espectral, ésta resulta poco atractiva frente a formulaciones espaciales como la empleada cuando la cavidad es eléctricamente grande respecto al circuito impreso. En este caso, se deben utilizar muchos modos para que la formulación espectral converja. Además, incluso con cavidades pequeñas el método neuronal de Chebychev consigue velocidades similares a los del método espectral. Hay que tener en cuenta que cada vez que se analice un nuevo circuito con la formulación espectral es necesario realizar un estudio sobre el número de modos que proporciona una solución estable. En general, dependiendo del tiempo invertido en la generación de las zonas y en el entrenamiento, existirá un número de análisis circuitales que convertirá en rentable al método neuronal basado en redes de Chebychev o en RBFNNs, siempre y cuando el método neuronal resultante sea más rápido que el método frente al que se compara.

El tamaño del conjunto de entrenamiento no está limitado en las redes de Chebychev. Por tanto, se han podido incluir en el espacio de entrada más coordenadas, como el grosor y permitividad relativa del dieléctrico o la longitud y anchura de la cavidad metálica. En las RBFNNs debido a limitaciones en la memoria RAM disponible era imposible aumentar la dimensión total del espacio de entrada. En cambio, las Redes neuronales de Chebychev han permitido la aproximación completa de las variaciones de las funciones de Green respecto a un espacio de entrada de dimensión siete. La precisión alcanzada es satisfactoria en todos los filtros analizados. La posibilidad de incluir las variaciones mencionadas en el método neuronal es muy interesante, ya que permite el análisis de un circuito determinado con dieléctricos diferentes o con el mismo dieléctrico en cavidades de distintos tamaños. Sin embargo, a pesar de la precisión y rapidez alcanzada, el método invierte un tiempo elevado en la división del espacio de entrada y en el entrenamiento de las redes neuronales. Además, no se consideró la existencia de frecuencias de resonancia en el ancho de banda de trabajo. Aunque éstas no aparezcan en la caja que sirve de base para determinar el rango de variación de las nuevas coordenadas, pueden surgir al variar las dimensiones de la cavidad. En este caso, habría que efectuar una densa división de la coordenada frecuencial, en torno a las nuevas frecuencias de resonancia, aumentando todavía más el tiempo de división del espacio de entrada.

Como se ha comentado, si se desean aproximar las funciones de Green en estructuras multicapa formadas por más de dos capas con metalizaciones, el método neuronal debe ser aplicado varias veces. Si la estructura consta de metalizaciones verticales, entonces la aplicación del método neuronal es más compleja. En este caso, se deben añadir dos nuevas coordenadas en la entrada de las redes neuronales (z, z'), por lo que el espacio de entrada constaría de siete coordenadas. Las RBFNNs difícilmente podrían aproximar las relaciones entre el espacio de salida (que se amplía a doce salidas) y un espacio de entrada de siete dimensiones. Se debería dividir el espacio de entrada en multitud de zonas para evitar problemas de saturación de la memoria. En cambio, como se ha visto en la tesis, las redes de Chebychev, entrenadas de forma novedosa, son capaces de trabajar con conjuntos de entre-

namiento de decenas de miles de muestras y de aproximar las funciones de Green con espacios de entrada de siete dimensiones. No obstante, el tiempo total de diseño de las redes neuronales puede disminuir el atractivo del método de análisis. Únicamente en el caso de realizar cientos de análisis se podría llegar a conseguir que el método neuronal fuera rentable. Más tarde se comenta uno de estos casos. Antes de abordar el problema de las metalizaciones verticales en estructuras apantalladas, sería conveniente e interesante emplear el método neuronal en estructuras no apantalladas con este tipo de metalizaciones. En este caso, la dimensión del espacio de entrada se reduciría a cuatro dimensiones: la distancia (ρ) entre el punto fuente y el punto observación, las coordenadas longitudinales (z, z') y la frecuencia. La aproximación de las funciones de Green en este tipo de estructuras no ha sido tratado y supondría un inicio para posteriores ampliaciones a estructuras apantalladas.

En cualquier caso, las Redes neuronales de Chebychev, entrenadas mediante los algoritmos desarrollados en esta tesis, deben ser seleccionadas para futuras aplicaciones del método neuronal por las ventajas que presentan frente a las RBFNNs: similar o superior capacidad de aproximación de funciones, mayor rapidez de entrenamiento, y sobre todo, gracias a la selección de conjuntos de neuronas muy reducidos. Además, no adolecen de los problemas de sobre-entrenamiento y limitación del conjunto de entrenamiento que padecen las RBFNNs. La única desventaja que presentan las Redes neuronales de Chebychev frente a las RBFNNs consiste en la limitación de la posición de los puntos de muestreo en el conjunto de entrenamiento. Esta limitación sería importante si no tuviéramos libertad para elegir los puntos de entrenamiento en el problema de aproximación. Por tanto, esta desventaja carece de relevancia en el problema de aproximación de las funciones de Green ya que tenemos total libertad para generar el conjunto de entrenamiento.

Por último, tenemos que referirnos a las ventajas de las Redes neuronales de Chebychev respecto a las RBFNNs en cuanto a la implantación en circuitos y chips hardware. Las redes neuronales poseen, en general, una estructura que permite una sencilla paralelización de las operaciones involucradas en el cálculo de las salidas. Esta característica posibilita la aceleración del mencionado cálculo neuronal, tanto en ordenadores con múltiples procesadores, como en dispositivos como las FPGAs. Hay que destacar que las Redes neuronales de Chebychev, entrenadas con el algoritmo OLS, son mucho más fácilmente programables en una FPGA que las RBFNNs a causa de diversas razones. En primer lugar, el espacio de trabajo de las funciones neuronales de las redes de Chebychev está acotado entre -1 y 1 . Este hecho permite que los números reales puedan ser expresados con precisión con un número limitado de bits. En segundo lugar, los valores de las salidas neuronales en las redes de Chebychev también están acotados entre -1 y 1 . Las salidas neuronales podrán ser asimismo expresadas con precisión. Por último, según todas las pruebas realizadas, el rango de variación de los pesos de las redes de Chebychev es mucho menor que el rango de variación de los pesos de las RBFNNs. De esta forma, dichos pesos se podrán representar de modo completo sin escatimar en el número de decimales, hecho que ocurriría si la parte entera del peso fuera muy grande.

Las novedosas herramientas de análisis han permitido el desarrollo de dos métodos de diseño de circuitos encapsulados multicapa. El primer método utiliza la técnica de los Algoritmos genéticos (AG). Ésta es una herramienta de optimización muy potente que posibilita la localización de extremos globales. Para ello, parte de una población de individuos cuyas prestaciones son mejoradas generación tras generación. La mejora se logra aplicando las estrategias encontradas en la naturaleza que permiten la mejora de las especies. En el caso del diseño de circuitos mediante los AG se parte

de un filtro inicial que no cumple las especificaciones. Con este filtro se forma una población inicial de individuos, es decir, de circuitos, que son alterados en sucesivas generaciones hasta que alguno cumple las especificaciones prefijadas. Existen pocas aplicaciones de los AG en el campo del diseño de circuitos encapsulados. Esto es debido a que los AG consumen mucho tiempo en su ejecución; éste es el precio que pagan por su capacidad de localización de extremos globales. Gracias al veloz método neuronal basado en RBFNNs, el AG ha sido aplicado al diseño de filtros encapsulados de forma práctica. Esta es una demostración de que, una vez entrenadas, las redes neuronales pueden ser empleadas en multitud de análisis circuitales. De este modo, el tiempo invertido en el diseño de todas las redes neuronales es compensado por la ganancia conseguida, haciendo rentable el método neuronal.

En la herramienta que utiliza el AG, se han seleccionado cuidadosamente aquellas técnicas que permiten una buena ejecución de este método de optimización. Uno de los puntos clave en la operación del AG es la definición de una adecuada función de rendimiento. Dicha función mide el grado de acercamiento de la solución que representa el individuo respecto al resultado buscado. Cada problema contiene peculiaridades que imponen la aplicación o creación de una función de rendimiento apropiada. En nuestro problema, se ha desarrollado una nueva función de rendimiento que mide, de forma eficiente, el grado de cumplimiento de las especificaciones por parte de un individuo (filtro). Una inadecuada definición de la función de rendimiento provocará problemas de convergencia en la operación del AG. Además, obviamente conducirá a soluciones erróneas que no satisfagan las especificaciones. El mayor problema que puede aparecer en la evolución del AG es la llamada convergencia prematura. Este problema consiste en la aparición en la población inicial de un individuo que posee un valor de rendimiento mucho mayor que el del resto de valores en la población. En ese caso todos los demás individuos tenderán a agruparse en torno al individuo preponderante. Con ello, el AG pierde la capacidad de exploración del espacio de solución y por tanto es incapaz de optimizar el filtro inicial. La nueva función de rendimiento penaliza el incumplimiento de las especificaciones y premia el cumplimiento sin que aparezcan individuos con valores excesivamente altos en el inicio del AG. Para ello, se utilizan funciones de tipo logístico. Este tipo de funciones tienden a un valor constante a partir de cierto valor de la variable independiente. De este modo, el valor de la función de rendimiento poseerá un valor máximo. El grado de penalización o gratificación del cumplimiento de las especificaciones se ajusta mediante los parámetros que controlan la pendiente de las funciones logísticas. En un principio, el ajuste de estos parámetros se realizó de forma heurística, gracias a la experiencia acumulada. Con el objetivo de evitar una inapropiada elección de los mencionados parámetros, se desarrolló un algoritmo que calcula los valores de forma automática. Concretamente, se calculan mediante un breve y sencillo proceso de optimización que tiene por objeto alcanzar cierta diferencia entre el valor de rendimiento de un filtro modelo y el rendimiento del filtro ideal que se desea obtener. El filtro modelo puede ser el filtro inicial o un filtro de la población inicial. La nueva función de rendimiento ha demostrado su habilidad en el diseño de varios filtros encapsulados. Gracias a ella, los AG convergen de forma robusta a una solución exitosa que cumple las especificaciones. La variable independiente en la función de rendimiento es la diferencia entre la respuesta del filtro y las especificaciones fijadas. Por tanto, la mencionada función del AG equivale a las funciones de error utilizadas en las técnicas de optimización por gradiente. Una de las líneas futuras consistirá en la aplicación de las funciones de error provenientes del campo de la optimización de circuitos mediante técnicas de gradiente en la técnica de los AG. De esta forma, se ampliará el estudio de la

convergencia de los AG con las nuevas funciones de rendimiento.

Para reducir en gran medida el tiempo de ejecución de los AG, en un futuro será necesario aplicar el método neuronal basado en Redes neuronales de Chebychev. De este modo, será posible reducir dicho tiempo en casi un orden de magnitud. Si se eligen los AG como método de diseño en una estructura multicapa apantallada formada por varias capas de dieléctrico con metalizaciones verticales, sería posible convertir en rentable el método neuronal en este tipo de cavidades. Como se ha comentado, sería necesario emplear un espacio de entrada de siete dimensiones por lo que se deberían usar casi obligatoriamente las Redes neuronales de Chebychev. Cuando el espacio de entrada aumenta hasta siete dimensiones el tiempo de división y entrenamiento de las redes neuronales alcanza varios cientos de horas. Si el AG se tuviera que utilizar en el diseño de varios circuitos, el tiempo ahorrado con el método neuronal frente a otras técnicas de análisis convertiría en rentable al método neuronal.

La segunda técnica de diseño de circuitos está fundamentada en la aproximación mediante redes neuronales de las relaciones existentes entre los parámetros eléctricos que caracterizan al circuito y las dimensiones físicas que los sintetizan. La técnica se ha empleado en filtros paso banda basados en resonadores con acoplos entre resonadores adyacentes, y sintonizados de modo síncrono. Tradicionalmente, en este tipo de filtros, el orden queda fijado tras seleccionar una función de transferencia (Chebychev, Butterworth) y unas especificaciones o máscara de atenuación. La función de transferencia permite hallar los coeficientes del prototipo paso bajo que sintetiza la respuesta deseada. El prototipo paso bajo conduce a la determinación de un filtro paso banda compuesto por elementos concentrados. Mediante operaciones sobre el filtro paso banda anterior se demuestra que los coeficientes de acoplo entre resonadores adyacentes, que determinan las interacciones entre resonadores, se pueden calcular a partir de los coeficientes del filtro paso bajo. Asimismo, los factores de calidad externos, por modelar el acoplo entre los puertos y los resonadores, pueden ser calculados con los mencionados coeficientes. El cálculo teórico de estos parámetros eléctricos, que en última instancia determinan el comportamiento de la respuesta eléctrica, es simple. El punto clave reside en la computación de las dimensiones que posibilitan que la interacción electromagnética entre resonadores corresponda a los parámetros eléctricos calculados. Un método para conseguir este objetivo consiste en la creación de tablas o curvas que relacionan los parámetros eléctricos y físicos de modo aproximado. En la presente tesis, se ha mostrado que las redes neuronales, en concreto la RBFNNs, son aptas para llevar a cabo esta tarea. Las redes neuronales presentan numerosas ventajas sobre los métodos basados en tablas o curvas. La principal ventaja reside en la dificultad de crear tablas o curvas cuando el número de parámetros crece. En este caso, hay que recurrir a curvas de nivel, por lo que incluso un diseñador experto dedicará un valioso tiempo a la determinación de la geometría final. En cambio, las redes neuronales son especialmente apropiadas para la aproximación de relaciones no lineales entre espacios multidimensionales. En cualquier caso, las redes neuronales calculan automáticamente la solución, evitando que el diseñador intervenga en el proceso. Además, son muy precisas y rápidas en el cálculo de los valores de salida. El esquema propuesto para el cálculo de las dimensiones físicas está compuesto por varias redes neuronales. Las salidas de unas redes pueden ser conectadas a las entradas de otras redes neuronales. Así ocurre en los ejemplos realizados, basados en resonadores formados por líneas acopladas y en forma de horquilla. Es un esquema muy flexible que puede adaptarse a muchas situaciones.

En este problema de aproximación se han empleado RBFNNs en lugar de Redes neuronales de Chebychev, principalmente por que los conjuntos de entrenamiento no son muy grandes (menos de 200 elementos). Además, el problema de aproximación no presenta las dificultades que existían en las funciones de Green, de modo que, aunque se ha empleado el algoritmo OLS, una estrategia como la de K-medias hubiera servido. La precisión alcanzada por las RBFNNs en esta tarea de aproximación ha sido muy elevada ya que se han conseguido niveles de error NMSE muy bajos. Los dos métodos neuronales desarrollados en esta tesis han sido empleados como simuladores electromagnéticos para generar los datos de entrenamiento y validación. A pesar de que con el método neuronal basado en RBFNNs se empleó la versión que utiliza la técnica de multiplicación por la distancia (ρ), éste presentó, en líneas generales, una buena precisión en el cálculo de los coeficientes de acoplo y los factores de calidad. El método neuronal basado en las redes de Chebychev utilizó la técnica de extracción de dos capas de imágenes espaciales. Las únicas diferencias entre ambos conjuntos de resultados se aprecian cuando la distancia entre resonadores se reduce, ya que provoca acoplos de caracterización más difícil. Hay que comentar diversos aspectos de la calidad del método propuesto. En los ejemplos mostrados, las dimensiones físicas calculadas por las RBFNNs conducen a filtros que son muy cercanos a la solución buscada, pero que no cumplen totalmente todas las especificaciones. En todos los casos es necesario aplicar un proceso de optimización al filtro obtenido con las RBFNNs. Gracias a la calidad de la respuesta, se pueden emplear técnicas basadas en el uso de gradiente sin temor a caer en mínimos locales. Por esta razón, y por la utilización del método neuronal de análisis, el proceso es breve y relativamente sencillo. Para mostrar la sencillez de la optimización se ha utilizado una de tipo directo cuando el método neuronal está basado en las Redes neuronales de Chebychev. La rapidez de este método de análisis permite que la tarea de optimización no sea muy pesada para el diseñador. Aunque siempre se debe intentar que el proceso sea automático, en este caso se deseaba demostrar que la aplicación de un conjunto de reglas optimiza el filtro en tan solo unas iteraciones. Por lo tanto, una de las líneas futuras surgidas de esta parte de la tesis consiste en la aplicación de la lógica difusa en la optimización de este tipo de filtros. Esta técnica es capaz de incorporar instrucciones como las que usamos los humanos en tareas cotidianas. Además, la lógica difusa puede generar un conjunto de instrucciones a partir de conjuntos de datos de entrada y salida. En este problema, es probable que consiguiera tardar poco en optimizar un filtro gracias al conocimiento incluido directamente, sin necesidad de entrenamiento, en el conjunto de instrucciones. Estas instrucciones, más las generadas por el entrenamiento con los datos de entrada y salida, formarían un conjunto muy completo de reglas difusas.

A pesar de que existe cierta inexactitud en la obtención de los parámetros eléctricos y cierto error en la aproximación de las RBFNNs, no se debe atribuir de forma principal a estas fuentes de imprecisión el hecho de que el filtro no cumpla las especificaciones totalmente. La principal causa de las diferencias entre la respuesta del filtro inicial y la respuesta ideal se debe a que la unión de todos los resonadores provoca la aparición de pequeños acoplos entre resonadores no adyacentes que alteran la respuesta del filtro. Estos acoplos no son tenidos en cuenta al caracterizar el acoplo entre dos resonadores adyacentes, ya que éste se estudia cuando ambos están aislados del resto de resonadores. Además, si el ancho de banda relativo es grande (entre un 10 % y un 20 %) los modelos circuitales que representan a los resonadores impresos pierden parte de su validez (aproximación de banda estrecha). Estos modelos son válidos alrededor de la frecuencia de resonancia y permiten la equiparación entre el filtro impreso y el filtro ideal formado por elementos concentrados. Sin embar-

go, lejos de la frecuencia de resonancia (frecuencia central del filtro) la equiparación pierde validez. Por ello, para estas frecuencias el coeficiente de acoplo (k) dejará de corresponder al valor del inversor de impedancias o admitancias del modelo ideal con elementos concentrados. Los filtros vistos en la bibliografía, diseñados a partir de tablas o curvas que relacionan parámetros eléctricos y dimensiones físicas, presentan el mismo problema. El filtro obtenido se encuentra cerca de la solución pero necesita de un proceso de optimización. Con el método neuronal desarrollado, el filtro inicial está tanto o más cerca de la solución buscada que los filtros iniciales obtenidos con curvas o tablas.

El método propuesto puede ser aplicado al diseño de filtros formados por resonadores que presenten acoplos cruzados (acoplos entre resonadores no adyacentes). En esta situación se utilizaría la misma metodología si las diferentes parejas de resonadores pudieran ser caracterizadas de forma independiente y aislada del resto de resonadores. Si alguna pareja de resonadores no se pudiera aislar, entonces la metodología debería extenderse para poder modelar los acoplos entre todos los resonadores involucrados y las dimensiones físicas pertinentes.

Asimismo, la metodología desarrollada puede aplicarse a resonadores asíncronos. En este caso, cada resonador posee una frecuencia de resonancia diferente por lo que todos los resonadores no resonarán a la frecuencia central del filtro. Para caracterizar los acoplos entre resonadores, la RBFNN debería tener como entradas dos frecuencias de resonancia diferentes y el coeficiente de acoplo. Las salidas de esta red serían las dos longitudes asociadas y la separación entre resonadores. La extensión a esta situación es simple. En este tipo de filtros el método desarrollado demostraría mejor su capacidad, ya que el espacio de entrada y salida tendría más dimensiones. En esta situación, el cálculo por medio de curvas o tablas es más difícil y engorroso. La asincronía en la resonancia puede servir para aplicar la metodología desarrollada a filtros transversales en los que varias resonancias no pudieran aislarse. En este caso, si los resonadores son asíncronos, éstos se pueden caracterizar de forma conjunta mediante la fase del coeficiente de reflexión. Los dos máximos en la derivada de la fase (retardo de grupo) indicarían las dos frecuencias de resonancia. A partir de la curva de la derivada de la fase se podrían extraer los dos acoplos, de modo que se podría entrenar a una red neuronal que relacionara estos parámetros eléctricos con la geometría de los resonadores. La utilización de la metodología desarrollada a filtros formados por resonadores que posean acoplos cruzados y a filtros con resonadores asíncronos constituye una de las líneas futuras de trabajo.

Los dos métodos de diseño propuestos siguen filosofías distintas. El Algoritmo genético parte de un filtro de poca calidad y lo optimiza hasta que cumpla las especificaciones. La segunda técnica busca directamente un filtro que cumpla las especificaciones. El tiempo empleado en la búsqueda de la solución se distribuye de diferente forma en ambos algoritmos. En el AG no se requiere tiempo inicial alguno y emplea un periodo muy largo en la optimización. En cambio, la técnica que usa RBFNNs necesita un periodo inicial prolongado pero consigue un filtro de tal calidad que el proceso de optimización es muy rápido. Según los ejemplos llevados a cabo, la técnica neuronal presenta ciertas ventajas sobre el AG. Este último requiere mucho tiempo de optimización en cada filtro si éste no se encuentra cerca de la solución. Por el contrario, una vez entrenadas las RBFNNs, éstas pueden ofrecer instantáneamente las dimensiones de multitud de filtros. Por supuesto, si la respuesta del filtro inicial del AG estuviera cerca de la solución la convergencia sería mucho más rápida. Sin embargo, en este caso una técnica que usara el gradiente sería suficiente. Por tanto, podemos concluir que si se tienen que diseñar varios filtros en una misma estructura fija y en un ancho de banda da-

do, la técnica basada en RBFNNs es más provechosa, ya que conforme diseñemos varios filtros el tiempo inicial invertido resultará despreciable. Si nos encontramos ante un difícil problema de optimización de un filtro complejo, entonces el AG, con una función de rendimiento como la definida en esta tesis, representa una técnica acertada. El AG explorará el espacio de solución durante varias generaciones hasta encontrar la mejor de las soluciones. Si estas soluciones no satisfacen totalmente las especificaciones, entonces éstas servirán como punto inicial para procesos de optimización basados en gradiente, como los aplicados en los filtros iniciales de la técnica basada en RBFNNs.

Apéndice

Apéndice A

Series infinitas de funciones sinusoidales

El método de suma por partes explicado en el Capítulo 2 es empleado para sumar de forma eficiente la serie modal de las funciones de Green en el dominio espectral cuando el punto observación se sitúa lejos del punto fuente. El algoritmo de suma por partes necesita de la evaluación de la suma de series infinitas sucesivas $(f_n^{(+1)}, f_n^{(+2)}, f_n^{(+3)}, \dots)$ o $(g_n^{(+1)}, g_n^{(+2)}, g_n^{(+3)}, \dots)$ basadas únicamente en funciones sinusoidales. Dichas funciones se pueden expresar en forma de funciones de tipo coseno para cada una de las funciones de Green como se aprecia en las ecuaciones (2.69, 2.70, 2.72 - 2.73, 2.74). Las series infinitas basadas en tipo coseno se calculan de forma analítica de forma que la convergencia de la serie total se verá acelerada significativamente. Para mostrar el proceso de cálculo que conduce a la obtención de la fórmula analítica de las series tipo coseno partimos de la siguiente serie:

$$f_n^{(+1)} = \cos\left(\frac{k\pi}{a}(x - x')\right) \quad (\text{A.1})$$

La función coseno se puede descomponer en una suma de funciones exponenciales complejas. La suma de una serie de funciones exponenciales se calcula de forma analítica, utilizando la suma de una progresión geométrica, como se observa en la siguiente ecuación

$$\begin{aligned} f_n^{(+2)} &= \sum_{k=n+1}^{\infty} \cos\left(\frac{k\pi}{a}(x - x')\right) = \frac{1}{2} \left(\sum_{k=n+1}^{\infty} e^{j\frac{k\pi}{a}(x-x')} + \sum_{k=n+1}^{\infty} e^{-j\frac{k\pi}{a}(x-x')} \right) \\ &= \frac{1}{2} \left(\frac{e^{j\frac{(n+1)\pi}{a}(x-x')}}{1 - e^{j\frac{\pi}{a}(x-x')}} + \frac{e^{-j\frac{(n+1)\pi}{a}(x-x')}}{1 - e^{-j\frac{\pi}{a}(x-x')}} \right) \\ &= \frac{-\sin\left(\frac{\pi}{a}\left(n + \frac{1}{2}\right)(x - x')\right)}{2 \sin\left(\frac{\pi}{2a}(x - x')\right)} = \frac{\cos\left(\frac{\pi}{a}\left(n + \frac{1}{2}\right)(x - x') + \frac{\pi}{2}\right)}{2 \sin\left(\frac{\pi}{2a}(x - x')\right)} \end{aligned} \quad (\text{A.2})$$

Volvemos a aplicar el mismo procedimiento a la siguiente serie infinita utilizando el resultado anterior:

$$\begin{aligned}
 f_n^{(+3)} &= \sum_{k=n+1}^{\infty} f_k^{(+2)} = \frac{-1}{2 \sin\left(\frac{\pi}{2a}(x-x')\right)} \sum_{k=n+1}^{\infty} \sin\left(\frac{\pi}{a}\left(k+\frac{1}{2}\right)(x-x')\right) \\
 &= \frac{-1}{4j \sin\left(\frac{\pi}{2a}(x-x')\right)} \left\{ \sum_{k=n+1}^{\infty} e^{j\frac{\pi}{a}\left(k+\frac{1}{2}\right)(x-x')} - \sum_{k=n+1}^{\infty} e^{-j\frac{\pi}{a}\left(k+\frac{1}{2}\right)(x-x')} \right\} \\
 &= \frac{-1}{4j \sin\left(\frac{\pi}{2a}(x-x')\right)} \left\{ e^{j\frac{\pi}{2a}(x-x')} \frac{e^{j\frac{\pi}{a}(n+1)(x-x')}}{1 - e^{j\frac{\pi}{a}(x-x')}} - e^{-j\frac{\pi}{2a}(x-x')} \frac{e^{-j\frac{\pi}{a}(n+1)(x-x')}}{1 - e^{-j\frac{\pi}{a}(x-x')}} \right\} \\
 &= \frac{-\cos\left(\frac{\pi}{a}(n+1)(x-x')\right)}{4 \sin^2\left(\frac{\pi}{a}(x-x')\right)} = \frac{\cos\left(\frac{\pi}{a}(n+1)(x-x') + \pi\right)}{4 \sin^2\left(\frac{\pi}{a}(x-x')\right)} \quad (\text{A.3})
 \end{aligned}$$

Para f_n^{+4} obtenemos:

$$f_n^{(+4)} = \frac{\sin\left(\frac{\pi}{a}\left(n+\frac{3}{2}\right)(x-x')\right)}{8 \sin^3\left(\frac{\pi}{a}(x-x')\right)} = \frac{\cos\left(\frac{\pi}{a}(n+1)(x-x') + \frac{3\pi}{2}\right)}{8 \sin^3\left(\frac{\pi}{a}(x-x')\right)} \quad (\text{A.4})$$

Con las expresiones previas se puede realizar la siguiente generalización:

$$\begin{aligned}
 f_n^{(+i)} &= \frac{-\sin\left(\frac{(i-2)\pi}{2} + \frac{\pi}{a}\left(n+\frac{i-1}{2}\right)(x-x')\right)}{2^{(i-1)} \left[\sin\left(\frac{x-x'}{2}\right)\right]^{(i-1)}} \\
 &= \frac{\cos\left(\frac{(i-1)\pi}{2} + \frac{\pi}{a}\left(n+\frac{i-1}{2}\right)(x-x')\right)}{2^{(i-1)} \left[\sin\left(\frac{x-x'}{2}\right)\right]^{(i-1)}} \quad \text{Para todo: } i = 2, 3, 4, \dots \quad (\text{A.5})
 \end{aligned}$$

Con estas funciones los términos involucrados en el resto, debidos a las funciones sinusoidales, se calculan de forma analítica acelerando su evaluación numérica. Generalmente con 5 – 10 términos en la serie del resto se pueden alcanzar precisiones de $10^{-5}\%$. La técnica de la suma por partes presenta una convergencia especialmente lenta cerca de la fuente. Esto es debido a que el índice inicial de la suma del resto $M = N$ (ver ecuación (2.63)) debe aumentar en esta zona para compensar la divergencia del numerador $1/\left(\sin\left(\frac{x-x'}{2}\right)\right)$ de la ecuación (A.5), cuando la distancia $(x-x')$ tiende a 0.

Apéndice B

Ortogonalidad de los regresores en la red neuronal de Chebychev en un espacio de entrada multidimensional

En este apéndice se demuestra el cumplimiento de la condición de ortogonalidad mostrada en la ecuación (4.33). En primer lugar, definimos el producto de dos regresores producidos por dos funciones neuronales y evaluados en el conjunto de datos de entrenamiento. Como se aprecia en la siguiente ecuación el producto total es igual a la suma de una serie de productos entre polinomios de Chebychev de distinto orden:

$$\mathbf{P}_i(\mathbf{x}_k)^T \mathbf{P}_j(\mathbf{x}_k) = \sum_{z=1}^N \left[T_{o_1(i)}(x_{k_1(z)}) T_{o_2(i)}(x_{k_2(z)}) \dots T_{o_n(i)}(x_{k_n(z)}) \right. \\ \left. T_{o_1(j)}(x_{k_1(z)}) T_{o_2(j)}(x_{k_2(z)}) \dots T_{o_n(j)}(x_{k_n(z)}) \right] \quad (\text{B.1})$$

Si desarrollamos el sumatorio de los N términos obtenemos:

$$\mathbf{P}_i(\mathbf{x}_k)^T \mathbf{P}_j(\mathbf{x}_k) = T_{o_1(i)}(x_{k_1(1)}) T_{o_2(i)}(x_{k_2(1)}) \dots T_{o_n(i)}(x_{k_n(1)}) T_{o_1(j)}(x_{k_1(1)}) T_{o_2(j)}(x_{k_2(1)}) \dots T_{o_n(j)}(x_{k_n(1)}) \\ + T_{o_1(i)}(x_{k_1(2)}) T_{o_2(i)}(x_{k_2(2)}) \dots T_{o_n(i)}(x_{k_n(2)}) T_{o_1(j)}(x_{k_1(2)}) T_{o_2(j)}(x_{k_2(2)}) \dots T_{o_n(j)}(x_{k_n(2)}) \\ \dots + T_{o_1(i)}(x_{k_1(N)}) T_{o_2(i)}(x_{k_2(N)}) \dots T_{o_n(i)}(x_{k_n(N)}) T_{o_1(j)}(x_{k_1(N)}) T_{o_2(j)}(x_{k_2(N)}) \dots T_{o_n(j)}(x_{k_n(N)}) \quad (\text{B.2})$$

Podemos agrupar los productos de cada término de la suma anterior en parejas de productos de polinomios. Cada pareja de polinomios corresponde a la misma coordenada:

$$\begin{aligned}
 \mathbf{P}_i(\mathbf{x}_k)^T \mathbf{P}_j(\mathbf{x}_k) = & \left[T_{o_1(i)}(x_{k_1(1)}) T_{o_1(j)}(x_{k_1(1)}) T_{o_2(i)}(x_{k_2(1)}) T_{o_2(j)}(x_{k_2(1)}) \dots T_{o_n(i)}(x_{k_n(1)}) T_{o_n(j)}(x_{k_n(1)}) \right] \\
 & + \left[T_{o_1(i)}(x_{k_1(2)}) T_{o_1(j)}(x_{k_1(2)}) T_{o_2(i)}(x_{k_2(2)}) T_{o_2(j)}(x_{k_2(2)}) \dots T_{o_n(i)}(x_{k_n(2)}) T_{o_n(j)}(x_{k_n(2)}) \right] \\
 & + \dots + \left[T_{o_1(i)}(x_{k_1(N)}) T_{o_1(j)}(x_{k_1(N)}) T_{o_2(i)}(x_{k_2(N)}) T_{o_2(j)}(x_{k_2(N)}) \dots T_{o_n(i)}(x_{k_n(N)}) T_{o_n(j)}(x_{k_n(N)}) \right]
 \end{aligned}
 \tag{B.3}$$

Si los regresores son diferentes, entonces existirá al menos una coordenada en la que los polinomios de Chebychev de la neurona i -ésima y la j -ésima sean diferentes. Suponemos que existe una única coordenada en la que aparecen polinomios diferentes y que dicha coordenada es la número n , es decir, $o_n(i) \neq o_n(j)$. Para comprobar que efectivamente el producto de los dos regresores es cero hay que definir la estructura del conjunto de datos. En las tablas B.1 y B.2 se expone una determinada forma de producir los N datos de entrenamiento. En este caso, se ha optado por generar $[p(1)p(2)p(3)p(4) \dots p(n-1)]$ bloques de $p(n)$ elementos cada uno. De esta manera, en cada bloque los índices de la coordenada n -ésima varían desde 1 hasta su valor final $p(n)$. Si los polinomios de la coordenada n -ésima fueran iguales y existiera otra coordenada m -ésima con polinomios diferentes, la estructura de datos se puede reordenar para obtener una representación similar a la expuesta en las tablas B.1 y B.2.

Tabla B.1: Estructura de los datos de entrenamiento: primeros $p(n-1)$ bloques de $p(n)$ elementos cada uno.

Índice (z)	$k_1(z)$	$k_2(z)$	\dots	$k_{n-1}(z)$	$k_n(z)$
1 ^{er} bloque					
1	1	1	\dots	1	1
2	1	1	\dots	1	2
\vdots	\vdots	\vdots	\ddots	\dots	\vdots
$p(n)$	1	1	\dots	1	$p(n)$
2 ^o bloque					
$p(n)+1$	1	1	\dots	2	1
$p(n)+2$	1	1	\dots	2	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$2 \cdot p(n)$	1	1	\dots	2	$p(n)$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$p(n-1)$ -ésimo bloque					
$[p(n-1)-1]p(n)+1$	1	1	\dots	$p(n-1)$	1
$[p(n-1)-1]p(n)+2$	1	1	\dots	$p(n-1)$	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$p(n-1)p(n)$	1	1	\dots	$p(n-1)$	$p(n)$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots

Tabla B.2: Estructura de los datos de entrenamiento: bloques correspondientes a la variación de las dos últimas coordenadas.

Índice (z)	$k_1(z)$	$k_2(z)$	\dots	$k_{n-1}(z)$	$k_n(z)$
$[p(3)p(4) \dots p(n-1)+1]$ -ésimo bloque					
$[p(3)p(4) \dots p(n-1)]p(n)+1$	1	2	\dots	1	1
$[p(3)p(4) \dots p(n-1)]p(n)+2$	1	2	\dots	1	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$[p(3)p(4) \dots p(n-1)+1]p(n)$	1	2	\dots	1	$p(n)$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$p(2)p(3)p(4) \dots p(n-1)$ -ésimo bloque					
$[p(2)p(3)p(4) \dots p(n-1)-1]p(n)+1$	1	$p(2)$	\dots	$p(n-1)$	1
$[p(2)p(3)p(4) \dots p(n-1)-1]p(n)+2$	1	$p(2)$	\dots	$p(n-1)$	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$[p(2)p(3)p(4) \dots p(n-1)]p(n)$	1	$p(2)$	\dots	$p(n-1)$	$p(n)$
$[p(2)p(3)p(4) \dots p(n-1)+1]$ -ésimo bloque					
$[p(2)p(3)p(4) \dots p(n-1)]p(n)+1$	2	1	\dots	1	1
$[p(2)p(3)p(4) \dots p(n-1)]p(n)+2$	2	1	\dots	1	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$[p(2)p(3)p(4) \dots p(n-1)+1]p(n)$	2	1	\dots	1	$p(n)$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$[p(1)p(2)p(3)p(4) \dots p(n-1)]$ -ésimo bloque					
$[p(1)p(2)p(3)p(4) \dots p(n-1)-1]p(n)+1$	$p(1)$	$p(2)$	\dots	$p(n-1)$	1
$[p(1)p(2)p(3)p(4) \dots p(n-1)-1]p(n)+2$	$p(1)$	$p(2)$	\dots	$p(n-1)$	2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$[p(1)p(2)p(3)p(4) \dots p(n-1)]p(n)$	$p(1)$	$p(2)$	\dots	$p(n-1)$	$p(n)$

La suma total de la ecuación (B.3) se puede expresar como la suma de los $p(1)p(2)\dots p(n-1)$ bloques de $p(n)$ elementos. Así, la suma correspondiente al primer bloque es:

$$\begin{aligned}
 1^{er} \text{ bloque} &= T_{o_1(i)}(x_{k_1(1)}) T_{o_1(j)}(x_{k_1(1)}) T_{o_2(i)}(x_{k_2(1)}) T_{o_2(j)}(x_{k_2(1)}) \dots T_{o_{n-1}(i)}(x_{k_{n-1}(1)}) T_{o_{n-1}(j)}(x_{k_{n-1}(1)}) \\
 &\cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \\
 &+ T_{o_1(i)}(x_{k_1(2)}) T_{o_1(j)}(x_{k_1(2)}) T_{o_2(i)}(x_{k_2(2)}) T_{o_2(j)}(x_{k_2(2)}) \dots T_{o_{n-1}(i)}(x_{k_{n-1}(2)}) T_{o_{n-1}(j)}(x_{k_{n-1}(2)}) \\
 &\cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 &+ \dots + T_{o_1(i)}(x_{k_1(p(n))}) T_{o_1(j)}(x_{k_1(p(n))}) T_{o_2(i)}(x_{k_2(p(n))}) T_{o_2(j)}(x_{k_2(p(n))}) \\
 &\dots T_{o_{n-1}(i)}(x_{k_{n-1}(p(n))}) T_{o_{n-1}(j)}(x_{k_{n-1}(p(n))}) \\
 &\cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right)
 \end{aligned} \tag{B.4}$$

En la Tabla B.1 observamos que en el primer bloque los $n - 1$ primeros elementos de cada producto son iguales para todos los términos desde $z = 1$ hasta $z = p(n)$. Por tanto, podemos expresar la suma total en este bloque como el producto del término común por la suma de los términos diferentes correspondientes a la última coordenada. Tomando $z = 1$ para los términos comunes obtenemos:

$$\begin{aligned}
 1^{er} \text{ bloque} &= T_{o_1(i)}(x_{k_1(1)}) T_{o_1(j)}(x_{k_1(1)}) T_{o_2(i)}(x_{k_2(1)}) T_{o_2(j)}(x_{k_2(1)}) \dots T_{o_{n-1}(i)}(x_{k_{n-1}(1)}) T_{o_{n-1}(j)}(x_{k_{n-1}(1)}) \\
 &\cdot \left[T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \right. \\
 &+ T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 &\left. + \dots + T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) \right]
 \end{aligned} \tag{B.5}$$

La suma de los términos del segundo bloque se define como:

$$\begin{aligned}
 2^o \text{bloque} = & T_{o_1(i)}(x_{k_1(p(n)+1)}) T_{o_1(j)}(x_{k_1(p(n)+1)}) T_{o_2(i)}(x_{k_2(p(n)+1)}) T_{o_2(j)}(x_{k_2(p(n)+1)}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(p(n)+1)}) T_{o_{n-1}(j)}(x_{k_{n-1}(p(n)+1)}) \\
 & \cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \\
 & + T_{o_1(i)}(x_{k_1(p(n)+2)}) T_{o_1(j)}(x_{k_1(p(n)+2)}) T_{o_2(i)}(x_{k_2(p(n)+2)}) T_{o_2(j)}(x_{k_2(p(n)+2)}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(p(n)+2)}) T_{o_{n-1}(j)}(x_{k_{n-1}(p(n)+2)}) \\
 & \cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 & + \dots + T_{o_1(i)}(x_{k_1(2p(n))}) T_{o_1(j)}(x_{k_1(2p(n))}) T_{o_2(i)}(x_{k_2(2p(n))}) T_{o_2(j)}(x_{k_2(2p(n))}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(2p(n))}) T_{o_{n-1}(j)}(x_{k_{n-1}(2p(n))}) \\
 & \cdot T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right)
 \end{aligned} \tag{B.6}$$

La operación realizada en (B.5) se aplica del mismo modo al segundo bloque. Tomando $z = p(n) + 1$ se obtiene la siguiente expresión:

$$\begin{aligned}
 2^o \text{bloque} = & T_{o_1(i)}(x_{k_1(p(n)+1)}) T_{o_1(j)}(x_{k_1(p(n)+1)}) T_{o_2(i)}(x_{k_2(p(n)+1)}) T_{o_2(j)}(x_{k_2(p(n)+1)}) \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(p(n)+1)}) T_{o_{n-1}(j)}(x_{k_{n-1}(p(n)+1)}) \\
 & \left[T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \right. \\
 & + T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 & \left. + \dots + T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) \right]
 \end{aligned} \tag{B.7}$$

De modo análogo la operación de agrupación se aplica al resto de bloques hasta llegar al último bloque:

$$\begin{aligned}
 \text{Último bloque} = & T_{o_1(i)}(x_{k_1(q+1)})T_{o_1(j)}(x_{k_1(q+1)})T_{o_2(i)}(x_{k_2(q+1)})T_{o_2(j)}(x_{k_2(q+1)}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(q+1)})T_{o_{n-1}(j)}(x_{k_{n-1}(q+1)}) \\
 & T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \\
 & + T_{o_1(i)}(x_{k_1(q+2)})T_{o_1(j)}(x_{k_1(q+2)})T_{o_2(i)}(x_{k_2(q+2)})T_{o_2(j)}(x_{k_2(q+2)}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(q+2)})T_{o_{n-1}(j)}(x_{k_{n-1}(q+2)}) \\
 & T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 & + \dots + T_{o_1(i)}(x_{k_1(q+p(n))})T_{o_1(j)}(x_{k_1(q+p(n))})T_{o_2(i)}(x_{k_2(q+p(n))})T_{o_2(j)}(x_{k_2(q+p(n))}) \dots \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(q+p(n))})T_{o_{n-1}(j)}(x_{k_{n-1}(q+p(n))}) \\
 & T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right)
 \end{aligned} \tag{B.8}$$

Tomando $q = [p(1)p(2)p(3)p(4) \dots p(n-1) - 1]p(n)$ y $z = [p(1)p(2)p(3)p(4) \dots p(n-1) - 1]p(n) + 1 = q + 1$ conseguimos la siguiente ecuación:

$$\begin{aligned}
 \text{Último bloque} = & T_{o_1(i)}(x_{k_1(q+1)})T_{o_1(j)}(x_{k_1(q+1)})T_{o_2(i)}(x_{k_2(q+1)})T_{o_2(j)}(x_{k_2(q+1)}) \\
 & \dots T_{o_{n-1}(i)}(x_{k_{n-1}(q+1)})T_{o_{n-1}(j)}(x_{k_{n-1}(q+1)}) \\
 & \left[T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \right. \\
 & + T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 & \left. + \dots + T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) \right]
 \end{aligned} \tag{B.9}$$

Debido a la propiedad de ortogonalidad de los polinomios de Chebychev el valor de la suma en todos los bloques es igual a cero. En efecto, como $T_{o_n(i)} \neq T_{o_n(j)}$, la suma en cada bloque de los términos mostrados en la ecuación (B.10) debe ser cero. Este hecho es debido a que: $o_n(i) \neq o_n(j)$, $o_n(i) < p(n)$, $o_n(j) < p(n)$ y a que el sumatorio se evalúa en las raíces del polinomio de orden $p(n)$:

$$\begin{aligned}
 & \left[T_{o_n(i)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(1-1/2)}{p(n)} \right) \right) \right. \\
 & + T_{o_n(i)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(2-1/2)}{p(n)} \right) \right) \\
 & \left. + \dots + T_{o_n(i)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) T_{o_n(j)} \left(\cos \left(\frac{\pi(p(n)-1/2)}{p(n)} \right) \right) \right] = 0 \quad \text{en todos los bloques}
 \end{aligned} \tag{B.10}$$

En la demostración previa se ha supuesto que existe un único polinomio diferente en una determinada coordenada. No obstante, si existen otras coordenadas en las que los polinomios de las dos neuronas son diferentes, se llega al mismo resultado. La agrupación en bloques opera de la misma forma y se llega así al mismo resultado.

Apéndice C

Aplicación del teorema de Stone-Weierstrass en las Redes neuronales de Chebychev

Para llevar a cabo la demostración de la capacidad de aproximación universal de las Redes neuronales de Chebychev recurriremos al teorema de Stone-Weierstrass. Su enunciado es el siguiente: si Z es un conjunto de funciones reales continuas en un conjunto compacto U y se cumplen las siguientes condiciones, [Rudin, 1976]:

1. Z es un álgebra, es decir, si Z es un conjunto cerrado para las operaciones de adición, multiplicación, y multiplicación escalar.
2. Z separa puntos en U , es decir, para todo $\mathbf{x}, \mathbf{y} \in U$ y $\mathbf{x} \neq \mathbf{y}$, entonces existe una función $f \in Z$ tal que $f(\mathbf{x}) \neq f(\mathbf{y})$.
3. Z no se anula en ningún punto \mathbf{x} de U , es decir, para cada \mathbf{x} de U existe una función $f \in Z$ tal que $f(\mathbf{x}) \neq 0$.

entonces el cierre uniforme de Z consiste en todas las funciones reales continuas en el conjunto compacto U ($C[U]$). También se dice que Z es denso U . En nuestro caso, el intervalo de trabajo $[-1, 1]$ en el caso unidimensional y $[-1, 1]^n$ en el caso multidimensional son conjuntos compactos ya que constituyen conjuntos cerrados y acotados, [Rudin, 1976].

El cierre uniforme de Z son todas aquellas funciones que son el límite de secuencias convergentes uniformemente de funciones que pertenecen a Z . En los aproximadores universales dicho cierre uniforme está formado por todas las funciones del conjunto compacto U . Por lo tanto, si se prueba que las Redes neuronales de Chebychev $f(\mathbf{x})$ constituyen un conjunto de funciones Z que cumpla las condiciones del teorema de Stone-Weierstrass entonces se puede asegurar que las Redes neuronales de Chebychev son un aproximador universal. De este modo se cumplirá:

$$\text{Dado } U = [-1, 1]^n \text{ para } \forall F(\mathbf{x}) \in C[U] \text{ y } \forall \epsilon, \exists f(\mathbf{x}) \in Z / |F(\mathbf{x}) - f(\mathbf{x})| < \epsilon \quad (\text{C.1})$$

Antes de empezar la demostración de la capacidad de las redes de Chebychev de aproximación universal obtendremos el resultado del producto de dos polinomios de Chebychev. Este resultado será muy útil, especialmente en la demostración de que el conjunto de Redes neuronales de Chebychev constituye un álgebra que será realizada en la próxima Sección. Así, partimos de la multiplicación de dos polinomios de Chebychev a la que aplicamos una relación trigonométrica:

$$\begin{aligned} T_i(x)T_j(x) &= \cos(i \arccos(x)) \cos(j \arccos(x)) \\ &= \frac{1}{2} [\cos((i+j) \arccos(x)) + \cos((i-j) \arccos(x))] \end{aligned} \quad (\text{C.2})$$

En el coseno de la resta de argumentos podemos aplicar:

$$\begin{aligned} \cos((i-j) \arccos(x)) &= \cos(-(j-i) \arccos(x)) \\ &= \cos((j-i) \arccos(x)) = \cos(|i-j| \arccos(x)) \end{aligned} \quad (\text{C.3})$$

Por lo tanto, introduciendo (C.3) en (C.2):

$$\begin{aligned} T_i(x)T_j(x) &= \frac{1}{2} [T_{i+j}(x) + T_{|i-j|}(x)] \\ &= \frac{1}{2} [T_s(x) + T_d(x)] \end{aligned} \quad (\text{C.4})$$

donde $s = i+j$ y $d = |i-j|$. De esta manera, expresamos el producto de dos polinomios de Chebychev como la suma de otros dos polinomios de Chebychev por el término $1/2$.

A continuación se demuestra el cumplimiento del teorema de Stone-Weierstrass para el caso del espacio de entrada unidimensional y multidimensional. Aunque la aplicación del teorema citado no sea necesaria en el caso unidimensional, se procede a su demostración ya que de esta forma se introduce una notación y procedimientos que serán útiles en la demostración de la capacidad de aproximación universal en el caso multidimensional.

C.0.1. Demostración del teorema de Stone-Weierstrass para un espacio de entrada unidimensional

Cada red neuronal de Chebychev con un espacio de entrada unidimensional genera una función diferente $f(x)$ en el conjunto compacto U de la ecuación (C.1) con $n = 1$. Las Redes neuronales de

Chebychev $f(x)$ constituyen un conjunto de funciones reales continuas llamado Z . En primer lugar se debe demostrar que Z es un álgebra en U . Para ello se requiere que Z sea un conjunto cerrado en la suma, la multiplicación y la multiplicación por una constante. Para que el conjunto Z sea cerrado en las anteriores operaciones, la función resultado de cualquiera de las tres operaciones mencionadas debe pertenecer a Z .

Empezaremos por la operación adición. Partimos de dos Redes neuronales de Chebychev diferentes:

$$f_1(x) = \sum_{i=1}^{N_1} \omega_i^{(1)} T_{o(i)}^{(1)}(x) \quad f_2(x) = \sum_{j=1}^{N_2} \omega_j^{(2)} T_{o(j)}^{(2)}(x) \quad (C.5)$$

donde $x \in [-1, 1]$ y N_1 y N_2 son el número de neuronas de la primera y segunda red neuronal respectivamente. No hay que confundir estos números con el número de datos de entrenamiento N . Cada red ha podido ser entrenada con diferentes conjuntos, número de datos y algoritmos de entrenamiento. Este hecho no afecta a las demostraciones posteriores. En general, $N_1 \neq N_2$. Existen tres grupos de polinomios en la suma: aquellos polinomios que solo pertenecen a la primera red neuronal, los polinomios propios de la segunda red neuronal, y los polinomios comunes a ambas redes neuronales.

$$\begin{aligned} f_1(x) + f_2(x) &= \sum_{i=1}^{N_1} \omega_i^{(1)} T_{o(i)}^{(1)}(x) + \sum_{j=1}^{N_2} \omega_j^{(2)} T_{o(j)}^{(1)}(x) \\ &= \sum_{q=1}^{N_q} \omega_q T_{o(q)}(x) + \sum_{p=1}^{N_p} \omega_p T_{o(p)}(x) + \sum_{u=1}^{N_u} \omega_u T_{o(u)}(x) \end{aligned} \quad (C.6)$$

donde los ω_q son los pesos de los polinomios propios de la primera red neuronal. De esta forma, $T_{o(q)}(x) = T_{o(i)}^{(1)}(x)$ si $o(i)$ es propio de la primera red neuronal. Asimismo, ω_p son los pesos de los polinomios propios de la segunda red neuronal. De esta forma $T_{o(p)}(x) = T_{o(j)}^{(2)}(x)$ si $o(j)$ es propio de la segunda red neuronal. Finalmente, los pesos ω_u son iguales a la suma $\omega_i + \omega_j$ si el polinomio se encuentra presente tanto en la primera red, como en la segunda ($o(i) = o(j)$). Podemos generar una nueva serie de términos de forma que se obtiene la siguiente relación:

$$\sum_{v=1}^{N_v} \omega_v T_{o(v)}(x) \quad (C.7)$$

$$\omega_v = \begin{cases} \omega_q, & 1 \leq v \leq N_q & 1 \leq q \leq N_q \\ \omega_p, & N_q + 1 \leq v \leq N_q + N_p & 1 \leq p \leq N_p \\ \omega_u, & N_q + N_p + 1 \leq v \leq N_q + N_p + N_u & 1 \leq u \leq N_u \end{cases} \quad (C.8)$$

La ecuación final (C.7) equivale a una red neuronal de Chebychev, es decir, a una función que pertenece a Z .

A continuación, se demuestra que la multiplicación entre dos Redes neuronales de Chebychev es una operación cerrada. La multiplicación de dos Redes neuronales de Chebychev distintas se expresa como:

$$\begin{aligned}
 f_1(x)f_2(x) &= \sum_{i=1}^{N_1} \omega_i^{(1)} T_{o(i)}^{(1)}(x) \sum_{j=1}^{N_2} \omega_j^{(2)} T_{o(j)}^{(2)}(x) \\
 &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} T_{o(i)}^{(1)}(x) T_{o(j)}^{(2)}(x) \\
 &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} \frac{1}{2} \left[T_{(o(i)+o(j))}(x) + T_{|o(i)-o(j)|}(x) \right] \\
 &= \frac{1}{2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} T_{(o(i)+o(j))}(x) + \frac{1}{2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} T_{|o(i)-o(j)|}(x) \quad (C.9)
 \end{aligned}$$

Esta multiplicación consta de dos grupos de $N_1 N_2$ polinomios de Chebychev. El primer grupo contiene los polinomios con orden suma de dos órdenes originales. El segundo contiene los polinomios cuyo orden es la diferencia en valor absoluto entre los órdenes de los polinomios originales. En los anteriores sumatorios aparecerán tres grupos distintos de polinomios. Por un lado, tendremos una serie de polinomios de igual orden. Estos polinomios pueden pertenecer únicamente al primer sumatorio, al segundo o a ambos. Supongamos que existen e órdenes diferentes que aparecen repetidos en dos o más polinomios del sumatorio total. Así, tendremos:

$$\begin{cases} N_{h(1)} = \text{Número total de polinomios de orden } h(1) \\ N_{h(2)} = \text{Número total de polinomios de orden } h(2) \\ \vdots \\ N_{h(e)} = \text{Número total de polinomios de orden } h(e) \end{cases} \quad (C.10)$$

Podemos agrupar los polinomios de mismo orden en sumatorios particulares. De este modo, el sumatorio de los polinomios de orden $h(u)$ donde $u \in 1, 2, \dots, e$ se representa como:

$$\sum_{k=1}^{N_{h(u)}} \omega_k^{h(u)} T_{h(u)}(x) = T_{h(u)}(x) \sum_{k=1}^{N_{h(u)}} \omega_k^{h(u)} \quad (C.11)$$

donde $\omega_k^{h(u)}$ es igual al producto de pesos originales $(1/2)\omega_i^{(1)}\omega_j^{(2)}$ cuyo polinomio de orden suma $(o(i) + o(j))$ o bien de orden resta $|o(i) - o(j)|$ es igual al orden $h(u)$.

Por otro lado, aparecerán polinomios formados bien por la suma de órdenes, bien por la diferencia de órdenes, que únicamente aparecerán una vez en todo el sumatorio. El número total de polinomios que aparecen en una única ocasión es:

$$N_d = 2N_1N_2 - \sum_{k=1}^e N_{h(k)} \quad (\text{C.12})$$

Estos polinomios también pueden ser agrupados en un sumatorio:

$$\sum_{k=1}^{N_d} \omega_k^d T_{o_k}^d(x) \quad (\text{C.13})$$

donde ω_k^d es igual al producto de pesos originales $(1/2)\omega_i^{(1)}\omega_j^{(2)}$ cuyo polinomio de orden suma $(o(i) + o(j))$ o bien de orden resta $|o(i) - o(j)|$ es igual al orden del polinomio $T_{o(k)}^d(x)$ que solo aparece una vez. Con la nueva ordenación de índices el sumatorio total se representa como:

$$\begin{aligned} f_1(x)f_2(x) &= T_{h(1)}(x) \sum_{k=1}^{N_{h(1)}} \omega_k^{h(1)} + T_{h(2)}(x) \sum_{k=1}^{N_{h(2)}} \omega_k^{h(2)} + \dots + T_{h(e)}(x) \sum_{k=1}^{N_{h(e)}} \omega_k^{h(e)} \\ &+ \sum_{k=1}^{N_d} \omega_k^d T_{o(k)}^d(x) = \sum_{v=1}^{e+N_d} \omega_v T_{o(v)}(x) \end{aligned} \quad (\text{C.14})$$

donde se ha realizado la siguiente asignación:

$$\omega_v = \begin{cases} \sum_{k=1}^{N_{h(1)}} \omega_k^{h(1)}, & v = 1 \\ \sum_{k=1}^{N_{h(2)}} \omega_k^{h(2)}, & v = 2 \\ \vdots & \\ \sum_{k=1}^{N_{h(e)}} \omega_k^{h(e)}, & v = e \\ \omega_k^d, & e + 1 \leq v \leq e + N_d \quad 1 \leq k \leq N_d \end{cases}$$

Los órdenes de los polinomios $T_{o(z)}(x)$ se asignan de la siguiente forma:

$$o(v) = \begin{cases} h(1) & v = 1 \\ h(2) & v = 2 \\ \vdots & \\ h(e) & v = e \\ o_k & e + 1 \leq v \leq e + N_d \quad 1 \leq k \leq N_d \end{cases} \quad (\text{C.15})$$

La red neuronal (C.14) equivale a una nueva red neuronal de Chebychev por lo que la función producto es una función que pertenece a Z .

A continuación, se muestra la demostración de que la operación consistente en la multiplicación de una red neuronal de Chebychev por una constante es una operación cerrada. Esta operación se muestra en la ecuación (C.16):

$$cf(x) = \sum_{i=1}^N \omega_i T_{o(i)}(x) = \sum_{i=1}^N \omega'_i T_{o(i)}(x) \quad (C.16)$$

donde $\omega'_i = c\omega_i$. De nuevo la función resultante pertenece al conjunto Z .

Al cumplir estas tres propiedades se demuestra que el conjunto de redes neuronales Z es un álgebra.

A continuación se debe demostrar que el conjunto Z separa puntos en U . Por tanto se debe cumplir que para $\forall x, y \in U$ con $x \neq y, \exists f \in Z / f(x) \neq f(y)$.

En primer lugar construimos la red neuronal $f(x) = \sum_{i=1}^N \omega_i T_{o(i)}(x)$ con $N = 1, \omega_1 = 1$ y $T_{o(1)}(x) = T_1(x) = x$. Por tanto $f(x)$ pertenece a Z . Esta red neuronal cumple la condición dada ya que $T_1(x) \neq T_1(y)$ si $x \neq y$.

Finalmente se debe demostrar que Z no se anula en ningún punto de U . Matemáticamente: $\forall x \in U, \exists f \in Z / f(x) \neq 0$.

Se puede construir fácilmente una función que siempre cumpla esta condición: $f(x) = \sum_{i=1}^N \omega_i T_{o(i)}(x)$ con $N = 1, \omega_1 = 1$, y $T_{o(1)}(x) = T_0(x) = 1$. Por tanto $f(x)$ pertenece a Z . Esta red neuronal cumple la condición dada, ya que $T_0(x) \neq 0$ debido a que $T_0(x)$ es igual a 1 para cualquier x . Concluimos que Z es un álgebra, separa puntos y no se anula en U y que por tanto las Redes neuronales de Chebychev en el caso de espacios de entrada unidimensionales son aproximadores universales.

C.0.2. Demostración del teorema de Stone-Weierstrass para un espacio de entrada multidimensional

Cada red neuronal de Chebychev con un espacio de entrada multidimensional genera una función diferente $f(x)$ en el conjunto compacto U de la ecuación (C.1) para cualquier valor de n . Las Redes neuronales de Chebychev $f(x)$ constituyen un conjunto de funciones reales continuas llamado Z . Al igual que en el caso unidimensional, demostraremos en primer lugar que Z es un álgebra en U .

Partimos de dos funciones diferentes que representan la operación de dos Redes neuronales de Chebychev:

$$f_1(\mathbf{x}) = \sum_{i=1}^{N_1} \omega_i^{(1)} P_i^{(1)}(\mathbf{x}) \quad (\text{C.17a})$$

$$f_2(\mathbf{x}) = \sum_{j=1}^{N_2} \omega_j^{(2)} P_j^{(2)}(\mathbf{x}) \quad (\text{C.17b})$$

donde $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ y $\mathbf{x} \in [-1, 1]^n$. Al igual que en el caso unidimensional, N_1 y N_2 son el número de neuronas de la primera y segunda red neuronal respectivamente. No hay que confundir estos números con el número de datos de entrenamiento N . De nuevo, cada red ha podido ser entrenada con diferentes conjuntos, número de datos y algoritmos de entrenamiento. Este hecho no afecta a las demostraciones posteriores. Cada función neuronal está compuesta por un productorio de polinomios de Chebychev:

$$P_i^{(1)}(\mathbf{x}) = T_{o_1(i)}^{(1)}(x_1) T_{o_2(i)}^{(1)}(x_2) \cdots T_{o_n(i)}^{(1)}(x_n) \quad (\text{C.18a})$$

$$P_j^{(2)}(\mathbf{x}) = T_{o_1(j)}^{(2)}(x_1) T_{o_2(j)}^{(2)}(x_2) \cdots T_{o_n(j)}^{(2)}(x_n) \quad (\text{C.18b})$$

La suma de ambas redes neuronales se expresa como:

$$\begin{aligned} f_1(\mathbf{x}) + f_2(\mathbf{x}) &= \sum_{i=1}^{N_1} \omega_i^{(1)} P_i^{(1)}(\mathbf{x}) + \sum_{j=1}^{N_2} \omega_j^{(2)} P_j^{(2)}(\mathbf{x}) \\ &= \sum_{i=1}^{N_1} \omega_i^{(1)} T_{o_1(i)}^{(1)}(x_1) T_{o_2(i)}^{(1)}(x_2) \cdots T_{o_n(i)}^{(1)}(x_n) \\ &\quad + \sum_{j=1}^{N_2} \omega_j^{(2)} T_{o_1(j)}^{(2)}(x_1) T_{o_2(j)}^{(2)}(x_2) \cdots T_{o_n(j)}^{(2)}(x_n) \end{aligned} \quad (\text{C.19})$$

Existen tres grupos de funciones neuronales en la suma: aquellas funciones que solo pertenecen a la primera red neuronal, las funciones neuronales propias de la segunda red neuronal, y finalmente las funciones neuronales comunes a ambas redes neuronales. Para que dos funciones neuronales sean iguales se deberá cumplir que ante un mismo valor de entrada \mathbf{x} se obtenga la misma salida. Para que esta condición se cumpla, el producto de polinomios debe ser el mismo en ambas funciones y además en el mismo orden. Así, se deberá cumplir que $o_k(i) = o_k(j)$ para $k = 1, 2, \dots, n$. Podemos reordenar los índices para conseguir los siguientes sumatorios:

$$\begin{aligned}
 f_1(\mathbf{x}) + f_2(\mathbf{x}) &= \sum_{q=1}^{N_q} \omega_q T_{o_1(q)}(x_1) T_{o_2(q)}(x_2) \cdots T_{o_n(q)}(x_n) \\
 &+ \sum_{p=1}^{N_p} \omega_p T_{o_1(p)}(x_1) T_{o_2(p)}(x_2) \cdots T_{o_n(p)}(x_n) \\
 &+ \sum_{u=1}^{N_u} \omega_u T_{o_1(u)}(x_1) T_{o_2(u)}(x_2) \cdots T_{o_n(u)}(x_n)
 \end{aligned} \tag{C.20}$$

donde los ω_q son los pesos de las funciones propias de la primera red neuronal. De esta forma $T_{o_k(q)}(x_k) = T_{o_k(i)}^{(1)}(x_k)$ para $k = 1, 2, \dots, n$ si la combinación de órdenes $[o_1(i), o_2(i), \dots, o_n(i)]$ es propia de la primera red neuronal. Asimismo, ω_p son los pesos de las funciones propias de la segunda red neuronal. De esta forma $T_{o_k(p)}(x_k) = T_{o_k(j)}^{(2)}(x_k)$ para $k = 1, 2, \dots, n$ si la combinación de órdenes $[o_1(j), o_2(j), \dots, o_n(j)]$ es propia de la segunda red neuronal. Finalmente, los pesos ω_u son iguales a la suma $\omega_i^{(1)} + \omega_j^{(2)}$ si la función neuronal se encuentra presente tanto en la primera red como en la segunda. En este caso se cumple: $[o_1(i) = o_1(j), o_2(i) = o_2(j), \dots, o_n(i) = o_n(j)]$. Podemos generar una nueva serie de términos de forma que se obtiene la siguiente relación:

$$\sum_{v=1}^{N_v} \omega_v P_v(\mathbf{x}) \tag{C.21}$$

Al igual que en la demostración de la suma cerrada en el caso unidimensional podemos formar la siguiente función:

$$\omega_v = \begin{cases} \omega_q, & 1 \leq v \leq N_q & 1 \leq q \leq N_q \\ \omega_p, & N_q + 1 \leq v \leq N_q + N_p & 1 \leq p \leq N_p \\ \omega_u, & N_q + N_p + 1 \leq v \leq N_q + N_p + N_u & 1 \leq u \leq N_u \end{cases} \tag{C.22}$$

La ecuación (C.21) equivale a una red neuronal de Chebychev, es decir, a una función que pertenece a Z .

Para demostrar que la multiplicación de dos funciones $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$ es una operación cerrada comenzamos con la definición de esta operación. Los productos de dos polinomios de Chebychev se desarrollan como suma de dos polinomios de Chebychev como se vio en la fórmula (C.4):

$$\begin{aligned}
 f_1(\mathbf{x})f_2(\mathbf{x}) &= \sum_{i=1}^{N_1} \omega_i^{(1)} P_i^{(1)}(\mathbf{x}) \sum_{j=1}^{N_2} \omega_j^{(2)} P_j^{(2)}(\mathbf{x}) \\
 &= \sum_{i=1}^{N_1} \omega_i^{(1)} T_{o_1(i)}^{(1)}(x_1) T_{o_2(i)}^{(1)}(x_2) \cdots T_{o_n(i)}^{(1)}(x_n) \sum_{j=1}^{N_2} \omega_j^{(2)} T_{o_1(j)}^{(2)}(x_1) T_{o_2(j)}^{(2)}(x_2) \cdots T_{o_n(j)}^{(2)}(x_n) \\
 &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} T_{o_1(i)}^{(1)}(x_1) T_{o_2(i)}^{(1)}(x_2) \cdots T_{o_n(i)}^{(1)}(x_n) T_{o_1(j)}^{(2)}(x_1) T_{o_2(j)}^{(2)}(x_2) \cdots T_{o_n(j)}^{(2)}(x_n) \\
 &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} T_{o_1(i)}^{(1)}(x_1) T_{o_1(j)}^{(2)}(x_1) T_{o_2(i)}^{(1)}(x_2) T_{o_2(j)}^{(2)}(x_2) \cdots T_{o_n(i)}^{(1)}(x_n) T_{o_n(j)}^{(2)}(x_n) \\
 &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \omega_i^{(1)} \omega_j^{(2)} \left[\frac{1}{2} \left(T_{o_1(i)+o_1(j)}(x_1) + T_{|o_1(i)-o_1(j)|}(x_1) \right) \right] \\
 &\quad + \left[\frac{1}{2} \left(T_{o_2(i)+o_2(j)}(x_2) + T_{|o_2(i)-o_2(j)|}(x_2) \right) \right] + \cdots \\
 &\quad + \left[\frac{1}{2} \left(T_{o_n(i)+o_n(j)}(x_n) + T_{|o_n(i)-o_n(j)|}(x_n) \right) \right]
 \end{aligned} \tag{C.23}$$

Desarrollando la serie de productos incluidos en el doble sumatorio:

$$\begin{aligned}
 f_1(\mathbf{x})f_2(\mathbf{x}) &= \sum_{i=1}^{N_1} \omega_i^{(1)} \omega_j^{(2)} \left[\left(\frac{1}{2} \right)^n T_{o_1(i)+o_1(j)}(x_1) T_{o_2(i)+o_2(j)}(x_2) \cdots T_{o_n(i)+o_n(j)}(x_n) \cdots \right. \\
 &\quad + \left(\frac{1}{2} \right)^n T_{o_1(i)+o_1(j)}(x_1) T_{|o_2(i)-o_2(j)|}(x_2) \cdots T_{o_n(i)+o_n(j)}(x_n) \cdots \\
 &\quad + \left(\frac{1}{2} \right)^n T_{o_1(i)+o_1(j)}(x_1) T_{o_2(i)+o_2(j)}(x_2) \cdots T_{|o_n(i)-o_n(j)|}(x_n) \cdots \\
 &\quad + \left(\frac{1}{2} \right)^n T_{o_1(i)+o_1(j)}(x_1) T_{|o_2(i)-o_2(j)|}(x_2) \cdots T_{|o_n(i)-o_n(j)|}(x_n) \cdots \\
 &\quad + \left(\frac{1}{2} \right)^n T_{|o_1(i)-o_1(j)|}(x_1) T_{o_2(i)+o_2(j)}(x_2) \cdots T_{o_n(i)+o_n(j)}(x_n) \cdots \\
 &\quad + \left(\frac{1}{2} \right)^n T_{|o_1(i)-o_1(j)|}(x_1) T_{|o_2(i)-o_2(j)|}(x_2) \cdots T_{o_n(i)+o_n(j)}(x_n) \cdots \\
 &\quad + \left(\frac{1}{2} \right)^n T_{|o_1(i)-o_1(j)|}(x_1) T_{o_2(i)+o_2(j)}(x_2) \cdots T_{|o_n(i)-o_n(j)|}(x_n) \cdots \\
 &\quad \left. + \left(\frac{1}{2} \right)^n T_{|o_1(i)-o_1(j)|}(x_1) T_{|o_2(i)-o_2(j)|}(x_2) \cdots T_{|o_n(i)-o_n(j)|}(x_n) \right]
 \end{aligned} \tag{C.24}$$

Se observa que cada uno de los términos del sumatorio situado entre corchetes equivale a una función neuronal. En efecto, cada término es un producto de n polinomios de Chebychev de forma que el polinomio k -ésimo toma como entrada el valor de la coordenada k -ésimo (x_k). El resultado final del doble sumatorio es la suma de una serie de funciones neuronales multiplicadas por los

pesos correspondientes $\omega_i^{(1)}\omega_j^{(2)}$ y por el término $(1/2)^n$. Al igual que en el caso unidimensional, existirán funciones neuronales repetidas entre todos los términos del doble sumatorio. Supongamos que existen e funciones distintas que se encuentran repetidas dos o más veces en la función final. El primer grupo de funciones iguales estará formada por los polinomios de Chebychev de órdenes $\mathbf{h}(1) = \{h_1(1), h_2(1), \dots, h_n(1)\}$. El segundo grupo de funciones tendrá los siguientes órdenes $\mathbf{h}(2) = \{h_1(2), h_2(2), \dots, h_n(2)\}$, y así sucesivamente hasta $\mathbf{h}(e)$. Cada grupo estará compuesto por un número determinado de funciones iguales:

$$\begin{cases} N_{h(1)} = \text{Número total de funciones con ordenes } \mathbf{h}(1) \\ N_{h(2)} = \text{Número total de funciones con ordenes } \mathbf{h}(2) \\ \vdots \\ N_{h(e)} = \text{Número total de funciones con ordenes } \mathbf{h}(e) \end{cases} \quad (\text{C.25})$$

Del mismo modo que en el caso unidimensional, podemos agrupar las funciones iguales en un sumatorio para obtener el peso de dicho grupo. De esta manera, el sumatorio de las funciones de órdenes $\mathbf{h}(u)$ donde $u \in 1, 2, \dots, e$ se representa como:

$$\sum_{k=1}^{N_{h(u)}} \omega_k^{h(u)} P_{h(u)}^h(\mathbf{x}) = P_{h(u)}^h(\mathbf{x}) \sum_{k=1}^{N_{h(u)}} \omega_k^{h(u)} \quad (\text{C.26})$$

donde $\omega_k^{h(u)}$ es igual al producto de $(\frac{1}{2})^n$ por los pesos originales $\omega_{i(1)}\omega_{j(2)}$ cuya función neuronal es fruto de la multiplicación de los polinomios de Chebychev que poseen los órdenes contenidos en \mathbf{h}_u .

El resto de funciones neuronales aparecerá una única vez en el sumatorio final. Como en el caso unidimensional existen N_d funciones de este tipo, ver (C.12).

Estas funciones también pueden ser agrupadas en un sumatorio:

$$\sum_{k=1}^{N_d} \omega_k^d P_{d(k)}^d(\mathbf{x}) \quad (\text{C.27})$$

donde ω_k^d es igual al producto de pesos originales $(\frac{1}{2})^n \omega_i^{(1)}\omega_j^{(2)}$ cuya función neuronal posee una serie de órdenes $\mathbf{d}(k) = \{d_1(k), d_2(k), \dots, d_n(k)\}$

Con la nueva ordenación de índices el sumatorio total se representa como:

$$\begin{aligned} f_1(\mathbf{x})f_2(\mathbf{x}) &= P_{h(1)}^h(\mathbf{x}) \sum_{k=1}^{N_{h(1)}} \omega_k^{h(1)} + P_{h(2)}^h(\mathbf{x}) \sum_{k=1}^{N_{h(2)}} \omega_k^{h(2)} + \dots + P_{h(e)}^h(\mathbf{x}) \sum_{k=1}^{N_{h(e)}} \omega_k^{h(e)} \\ &+ \sum_{k=1}^{N_d} \omega_k^d P_{d(k)}^d(\mathbf{x}) = \sum_{v=1}^{e+N_d} \omega_v P_v(\mathbf{x}) \end{aligned} \quad (\text{C.28})$$

donde se ha realizado la siguiente asignación:

$$\omega_v = \begin{cases} \sum_{k=1}^{N_{h(1)}} \omega_k^{h(1)}, & v = 1 \\ \sum_{k=1}^{N_{h(2)}} \omega_k^{h(2)}, & v = 2 \\ \vdots & \\ \sum_{k=1}^{N_{h(e)}} \omega_k^{h(e)}, & v = e \\ \omega_k^d, & e + 1 \leq v \leq e + N_d \quad 1 \leq k \leq N_d \end{cases}$$

Los órdenes de las polinomios de Chebychev que componen las funciones neuronales se asignan de la siguiente forma:

$$\mathbf{o}(v) = \{o_1(v), o_2(v), \dots, o_n(v)\} = \begin{cases} \mathbf{h}(1) & v = 1 \\ \mathbf{h}(2) & v = 2 \\ \vdots & \\ \mathbf{h}(e) & v = e \\ \mathbf{d}(k) & e + 1 \leq v \leq e + N_d \quad 1 \leq k \leq N_d \end{cases} \quad (\text{C.29})$$

La función de la ecuación (C.28) equivale a una nueva red neuronal de Chebychev por lo que la función producto de dos redes neuronales pertenece a Z .

Seguidamente, demostraremos que la multiplicación de una red neuronal por una constante produce una función que pertenece al conjunto de redes neuronales, es decir, a Z :

$$cf(\mathbf{x}) = \sum_{i=1}^N \omega_i P_i(\mathbf{x}) = \sum_{i=1}^N \omega'_i P_i(\mathbf{x}) \quad (\text{C.30})$$

donde $\omega'_i = c\omega_i$. Con este cambio en los pesos de la capa lineal se demuestra que la función resultante pertenece al conjunto Z .

Al cumplir estas tres propiedades se demuestra que Z es un álgebra.

Para continuar con la demostración del cumplimiento del teorema de Stone-Weierstrass se debe probar que el conjunto Z separa puntos en U . Matemáticamente, la operación se expresa de la siguiente forma: $\forall \mathbf{x}, \mathbf{y} \in U$ con $\mathbf{x} \neq \mathbf{y}$, $\exists f \in Z / f(\mathbf{x}) \neq f(\mathbf{y})$.

Si $\mathbf{x} \neq \mathbf{y}$, entonces existe al menos una coordenada en la que se cumple $x_j \neq y_j$. Construimos la siguiente red neuronal compuesta por una única función neuronal $f(\mathbf{x}) = \omega_1 P_1(\mathbf{x})$ que pertenece a Z :

$$\omega_1 = 1 \quad (C.31a)$$

$$P_1(\mathbf{x}) = T_{o_1(1)}(x_1)T_{o_2(1)}(x_2) \dots T_{o_n(1)}(x_n) \quad (C.31b)$$

$$T_{o_j(1)} = T_1 \quad T_{o_1(1)} = T_{o_2(1)} = \dots = T_{o_n(1)} = T_0 = 1 \quad (C.31c)$$

Esta red neuronal cumple la condición dada ya que $f(\mathbf{x}) = x_j \neq f(\mathbf{y}) = y_j$ debido a que $x_j \neq y_j$.

Finalmente, hay que demostrar que el conjunto Z no se anula en ningún punto de U . Matemáticamente, se debe probar que: $\forall \mathbf{x} \in U, \exists f \in Z / f(\mathbf{x}) \neq 0$.

Se puede construir una función que siempre cumpla esta condición y que pertenezca a Z : $f(\mathbf{x}) = \sum_{i=1}^N \omega_i P_i(\mathbf{x})$ con $N = 1$:

$$\omega_1 = 1 \quad (C.32a)$$

$$P_1(\mathbf{x}) = T_{o_1(1)}(x_1)T_{o_2(1)}(x_2) \dots T_{o_n(1)}(x_n) \quad (C.32b)$$

$$T_{o_1(1)} = T_{o_2(1)} = \dots = T_{o_n(1)} = T_1 = 1 \quad (C.32c)$$

Esta red neuronal cumple la condición dada ya que $f(\mathbf{x}) = 1$ para cualquier \mathbf{x} .

Gracias a que la red de Chebychev para el caso multidimensional cumple las condiciones enumeradas en el teorema de Stone-Weierstrass podemos concluir que constituye un aproximador universal en el conjunto compacto $[-1, 1]^n$.

Apéndice D

Zonas fuente en el método neuronal basado en Redes neuronales de Chebychev

A continuación se muestran las zonas fuente de la cavidad marcada con el número 1 en la Figura 4.18 cuando se emplean las Redes neuronales de Chebychev. esta cavidad coincide con la caja dibujada en 3.37 y 3.38. Todas las Tablas de este apéndice no han sido incluidas en el Capítulo 4 por motivos de espacio y comodidad de lectura.

Tabla D.1: Zonas fuente correspondientes a la primera zona observación de la región MS.

	1ª zona observ.	
Zona	Inf.	Sup.
1ª zona	0.30000	0.50000
2ª zona	0.16667	0.30000
3ª zona	0.03333	0.16667
4ª zona	0.00000	0.03333

Tabla D.2: Zonas fuente correspondientes a la segunda zona observación de la región MS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.3000	0.5000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.1667	0.3000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.0333	0.1667	0.06667	0.16667	0.03333	0.16667	0.03333	0.16667
4ª zona	0.0083	0.0333	0.02868	0.06667	0.00000	0.03333	0.01591	0.03333
5ª zona	0.0000	0.0083	0.01591	0.02868			0.00000	0.01591
6ª zona			0.00000	0.01591				

Tabla D.3: Zonas fuente correspondientes a la tercera zona observación de la región MS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.06667	0.16667	0.07158	0.16667	0.03333	0.16667	0.07158	0.16667
4ª zona	0.03854	0.06667	0.03978	0.07158	0.00000	0.03333	0.04781	0.07158
5ª zona	0.01745	0.03854	0.00000	0.03978			0.03978	0.04781
6ª zona	0.00000	0.01745					0.00000	0.03978

Tabla D.4: Zonas fuente correspondientes a la cuarta zona observación de la región MS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.08054	0.16667	0.03333	0.16667	0.08054	0.16667
4ª zona	0.00833	0.03333	0.05172	0.08054	0.00000	0.03333	0.05172	0.08054
5ª zona	0.00000	0.00833	0.00000	0.05172			0.00000	0.05172

Tabla D.5: Zonas fuente correspondientes a la primera zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.08951	0.16667	0.03333	0.16667	0.06366	0.16667
4ª zona	0.00833	0.03333	0.06366	0.08951	0.00000	0.03333	0.00000	0.06366
5ª zona	0.00208	0.00833	0.00000	0.06366				
6ª zona	0.00000	0.00208						

Tabla D.6: Zonas fuente correspondientes a la segunda zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.11336	0.16667	0.03333	0.16667	0.09549	0.16667
4ª zona	0.00833	0.03333	0.09549	0.11336	0.00000	0.03333	0.00000	0.09549
5ª zona	0.00000	0.00833	0.00000	0.09549				

Tabla D.7: Zonas fuente correspondientes a la tercera zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.14320	0.16667	0.03333	0.16667	0.14320	0.16667
4ª zona	0.00833	0.03333	0.00000	0.14320	0.00000	0.03333	0.00000	0.14320
5ª zona	0.00000	0.00833						

A continuación se muestran las zonas fuente de la cavidad marcada con el número 2 en la Figura 4.18 cuando se emplean las Redes neuronales de Chebychev. Esta cavidad coincide con la caja marcada con un 2 en 3.53.

A continuación se muestran las zonas fuente de la cavidad dibujada en las Figuras 4.45 y 4.48.

Tabla D.8: Zonas fuente correspondientes a la cuarta zona observación de la región DS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.19090	0.30000	0.16667	0.30000	0.19090	0.30000
3ª zona	0.03333	0.16667	0.00000	0.19090	0.03333	0.16667	0.00000	0.19090
4ª zona	0.00000	0.03333			0.00000	0.03333		

Tabla D.9: Zonas fuente correspondientes a la quinta zona observación de la región DS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.25460	0.30000	0.16667	0.30000	0.25460	0.30000
3ª zona	0.03333	0.16667	0.00000	0.25460	0.03333	0.16667	0.00000	0.25460
4ª zona	0.00000	0.03333			0.00000	0.03333		

Tabla D.10: Zonas fuente correspondientes a la primera zona observación de la región MS.

Zona	1ª zona observ.	
	Inf.	Sup.
1ª zona	0.30000	0.50000
2ª zona	0.16667	0.30000
3ª zona	0.03333	0.16667
4ª zona	0.00000	0.03333

Tabla D.11: Zonas fuente correspondientes a la segunda zona observación de la región MS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.03333	0.16667	0.03333	0.16667	0.03333	0.16667
4ª zona	0.00000	0.03333	0.01591	0.03333	0.00000	0.03333	0.01591	0.03333
5ª zona			0.00000	0.01591			0.00000	0.01591

Tabla D.12: Zonas fuente correspondientes a la tercera zona observación de la región MS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.03978	0.16667	0.03333	0.16667	0.03978	0.16667
4ª zona	0.00000	0.03333	0.00000	0.03978	0.00000	0.03333	0.00000	0.03978

Tabla D.13: Zonas fuente correspondientes a la cuarta zona observación de la región MS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.05172	0.16667	0.03333	0.16667	0.05172	0.16667
4ª zona	0.00000	0.03333	0.00000	0.05172	0.00000	0.03333	0.00000	0.05172

Tabla D.14: Zonas fuente correspondientes a la primera zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.08951	0.16667	0.03333	0.16667	0.06366	0.16667
4ª zona	0.01458	0.03333	0.06366	0.08951	0.00000	0.03333	0.00000	0.06366
5ª zona	0.00638	0.01458	0.00000	0.06366				
6ª zona	0.00160	0.00638						
7ª zona	0.00000	0.00160						

Tabla D.15: Zonas fuente correspondientes a la segunda zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.14320	0.16667	0.03333	0.16667	0.14320	0.16667
4ª zona	0.00833	0.03333	0.00000	0.14320	0.00833	0.03333	0.00000	0.14320
5ª zona	0.00000	0.00833			0.00000	0.00833		

Tabla D.16: Zonas fuente correspondientes a la tercera zona observación de la región DS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.21831	0.30000	0.16667	0.30000	0.19090	0.30000
3ª zona	0.03333	0.16667	0.19090	0.21831	0.03333	0.16667	0.00000	0.19090
4ª zona	0.00833	0.03333	0.00000	0.19090	0.00000	0.03333		
5ª zona	0.00000	0.00833						

Tabla D.17: Zonas fuente correspondientes a la cuarta zona observación de la región DS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.23870	0.30000	0.16667	0.30000	0.23870	0.30000
3ª zona	0.03333	0.16667	0.00000	0.23870	0.03333	0.16667	0.00000	0.23870
4ª zona	0.00833	0.03333			0.00000	0.03333		
5ª zona	0.00208	0.00833						
6ª zona	0.00000	0.00000						

Tabla D.18: Zonas fuente correspondientes a la quinta zona observación de la región DS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30230	0.50000	0.30000	0.50000	0.30230	0.50000
2ª zona	0.17083	0.30000	0.00000	0.30230	0.17083	0.30000	0.00000	0.30230
3ª zona	0.03917	0.17083			0.03917	0.17083		
4ª zona	0.00000	0.03917			0.00000	0.03917		

Tabla D.19: Zonas fuente correspondientes a la primera zona observación de la región MS.

Zona	1ª zona observ.	
	Inf.	Sup.
1ª zona	0.30000	0.50000
2ª zona	0.16667	0.30000
3ª zona	0.03333	0.16667
4ª zona	0.00000	0.03333

Tabla D.20: Zonas fuente correspondientes a la segunda zona observación de la región MS.

Zona	Arriba		Abajo		Derecha		Izquierda	
	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.03333	0.16667	0.03333	0.16667	0.03333	0.16667
4ª zona	0.00000	0.03333	0.01590	0.03333	0.00000	0.03333	0.01590	0.03333
5ª zona			0.00000	0.01590			0.00000	0.01590

Tabla D.21: Zonas fuente correspondientes a la primera zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.06667	0.16667	0.06667	0.16667	0.03333	0.16667	0.06667	0.16667
4ª zona	0.00000	0.06667	0.03190	0.06667	0.00000	0.03333	0.03190	0.06667
5ª zona			0.00000	0.03190			0.00000	0.03190

Tabla D.22: Zonas fuente correspondientes a la segunda zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.08949	0.16667	0.03333	0.16667	0.06370	0.16667
4ª zona	0.00000	0.03333	0.06370	0.08949	0.00000	0.03333	0.00000	0.06370
5ª zona			0.00000	0.06370				

Tabla D.23: Zonas fuente correspondientes a la tercer zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.11140	0.16667	0.03333	0.16667	0.11140	0.16667
4ª zona	0.00000	0.03333	0.00000	0.11140	0.00000	0.03333	0.00000	0.11140

Tabla D.24: Zonas fuente correspondientes a la cuarta zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000	0.16667	0.30000
3ª zona	0.03333	0.16667	0.15920	0.16667	0.03333	0.16667	0.15920	0.16667
4ª zona	0.00000	0.03333	0.00000	0.15920	0.00000	0.03333	0.00000	0.15920

Tabla D.25: Zonas fuente correspondientes a la quinta zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.18310	0.30000	0.16667	0.30000	0.18310	0.30000
3ª zona	0.03333	0.16667	0.00000	0.18310	0.03333	0.16667	0.00000	0.18310
4ª zona	0.00000	0.03333			0.00000	0.03333		

Tabla D.26: Zonas fuente correspondientes a la sexta zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.22727	0.30000	0.16667	0.30000	0.20290	0.30000
3ª zona	0.03333	0.16667	0.20290	0.22727	0.03333	0.16667	0.00000	0.20290
4ª zona	0.00000	0.03333	0.00000	0.20290	0.00000	0.03333		

Tabla D.27: Zonas fuente correspondientes a la séptima zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.22280	0.30000	0.16667	0.30000	0.22280	0.30000
3ª zona	0.03333	0.16667	0.00000	0.22280	0.03333	0.16667	0.00000	0.22280
4ª zona	0.00000	0.03333			0.00000	0.03333		

Tabla D.28: Zonas fuente correspondientes a la octava zona observación de la región DS.

	Arriba		Abajo		Derecha		Izquierda	
Zona	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
1ª zona	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000	0.30000	0.50000
2ª zona	0.16667	0.30000	0.27060	0.30000	0.16667	0.30000	0.27060	0.30000
3ª zona	0.03333	0.16667	0.00000	0.27060	0.03333	0.16667	0.00000	0.27060
4ª zona	0.00000	0.03333			0.00000	0.03333		

Índice alfabético

- ADS, 225, 248, 275, 276
- Algoritmo de división del espacio de entrada,
 - 129, 203
 - varianza adaptativa, 121
 - zonas frecuenciales, 122
 - zonas fuente, 122, 133
 - zonas observación, 122, 133
 - varianza fija, 103, 113
 - zonas frecuenciales, 106
 - zonas fuente, 106, 113, 115
 - zonas observación, 105, 114
 - zona de altura del dieléctrico, 231
 - zona de anchura del dieléctrico, 233
 - zona de longitud del dieléctrico, 233
 - zona de permitividad del dieléctrico, 231
 - zona de transición entre zonas circulares y cuadradas, 130
 - zonas circulares, 130, 138
 - zonas cuadradas, 103, 130, 138, 200, 249, 268
 - zonas fuente, 202
 - zonas observación, 201
- Algoritmo de entrenamiento no supervisado, 71–73
- Algoritmo de entrenamiento supervisado, 71, 74, 81
- Algoritmo de optimización minimax, 275
- Algoritmo genético (AG), 243–260
 - convergencia prematura, 245, 258
 - cromosoma, 243
 - definición, 243
 - elitismo, 245, 258
 - funciones logísticas, 246
 - gen, 243, 251, 257
 - mutación, 245, 251
 - nueva función de rendimiento, 246, 251, 254
 - parámetro z de control de la pendiente en la función de rendimiento, 247, 254
 - población, 244, 250, 257
 - punto de cruce, 245, 250
 - selección, 244
 - técnica de la ruleta, 244
 - técnica del torneo, 245
- Aproximación universal, 62, 68, 69, 176, 186, 191
- Capa de influencia de imágenes, 33, 129
- Cavidad apantallada, 86
- Coefficiente de acoplo, 262, 266, 270, 272, 281
- Conjunto de imágenes básico, 32
- Constante de propagación en una línea de transmisión en el modelo circuital equivalente, 28
- Distancia de “tapping”, 268, 271, 288
- Distancia entre el punto observación y el punto fuente (ρ), 82, 129
- Ecuación integral (IE), 23, 111, 112, 129
- Error NMSE, 91, 113, 118, 122, 144, 192, 215
- Estructura multicapa, 2, 40, 82, 116, 137
- Factor de calidad externo, 262, 263, 265, 270
- Factor de Neumann, 28
- Factor de normalización del potencial eléctrico escalar, 27, 34, 36
- Factor de normalización del potencial magnético vectorial, 27, 38
- FPGA, 230
- Frecuencia de resonancia, 262, 263, 265, 270
- Funciones de diferencias:
 - caso bidimensional, 45
 - caso unidimensional, 41, 42
- Funciones de Green de los potenciales, 43, 80
 - formulación espacial, 25
 - parte cuasi-estática, 34, 47

- parte dinámica, 34
- serie de imágenes, 32
- serie espectral, 41
- Función de Green del potencial eléctrico escalar, 25, 31, 32, 37, 46, 50, 92, 94, 130
 - parte cuasi-estática, 34
 - parte dinámica, 34
- Función de Green del potencial magnético vectorial, 25, 32, 38, 39, 46
- Generador de intensidad que representa una excitación puntual en el modelo circuital equivalente, 28, 35
- Golden section search, 269
- Impedancia característica de una línea de transmisión del modelo circuital equivalente, 28, 35
- Impedancia de entrada de una línea de transmisión del modelo circuital equivalente, 28
- Intensidad de una línea de transmisión del modelo circuital equivalente, 28, 30
- LAPACK, 109
- Longitud del resonador, 267–269, 271, 281, 288
- MATLAB, 97, 98, 100, 120, 255
- Modelo circuital equivalente de la estructura multicapa apantallada, 26, 28
- Modo transversal eléctrico $TE_{m,n}$, 26
- Modo transversal magnético $TM_{m,n}$, 26
- Máquinas de vectores soporte (SVM), 63, 97, 100
 - Dimensión de Vapnik-Chervonenkis (VC), 63
 - Función de kernel, 64
- Método de análisis espectral, 222
- Método de Gram-Schmidt, 77
- Método de los momentos (MoM), 58, 111, 112, 118
- Números de onda transversal, 26
- Ortogonalidad discreta, 173, 183
- Perceptrón multicapa (MLP), 61, 97
 - función de activación neuronal, 61
- Polinomio minimax, 187
- Polinomios de Chebychev, 167
 - definición, 174
 - equioscilaación, 187
 - fórmula de aproximación, 175
 - fórmula trigonométrica, 174
 - ortogonalidad discreta, 175
 - raíces, 175
- Polinomios de Hermite, 173, 198
- Polinomios de Legendre, 174
- Redes neuronales de Chebychev, 167–214, 280
 - algoritmo OLS
 - carga computacional, 189
 - entrada multidimensional, 181
 - matriz de regresores, 178, 179
 - pesos lineales, 180
 - ratio de reducción del error, 177, 179, 184, 185, 194
 - regresor, 179
 - salida multidimensional, 184
 - simplificaciones, 180
 - anulación en ningún punto: caso multidimensional, 332
 - anulación en ningún punto: caso unidimensional, 326
 - estructura, 178
 - función neuronal, 177, 181
 - multiplicación cerrada: caso multidimensional, 328
 - multiplicación cerrada: caso unidimensional, 324
 - multiplicación de polinomios, 181, 322
 - multiplicación por una constante cerrada: caso unidimensional, 325
 - separación de puntos: caso multidimensional, 331
 - separación de puntos: caso unidimensional, 326
 - sin algoritmo OLS, 192
 - suma cerrada: caso multidimensional, 326
 - suma cerrada: caso unidimensional, 323
- Redes neuronales de funciones de base radial (RBFNN), 60, 65–136, 248, 262, 268

- ul style="list-style-type: none; padding-left: 0;">
- algoritmo de agrupación en K-medias ("K-means"), 73, 91, 102
- algoritmo FOLS, 134
- algoritmo OLS, 74, 91
 - eficiencia, 170
 - energía de salida, 77, 80
 - matrix de regresores, 75
 - matriz de productos de regresores y vectores salida, 135
 - matriz de regresores, 68, 72, 74, 96, 102
 - matriz de regresores ortogonales, 76
 - pesos lineales, 75, 80, 90
 - ratio de reducción del error, 78, 79, 90, 170
 - regresor, 75, 78, 89, 134, 170
 - regresor ortogonal, 75, 78, 89
 - salida multidimensional, 80
 - simplificaciones, 89
 - versión mejorada, 134
- algoritmos de entrenamiento, 71
- bias o constante(b), 69
- definición, 65
- estructura, 69
- estructura en la aproximación de las funciones de Green, 88
- función de base radial, 67, 69
- función gaussiana, 70
- selección de centros aleatoria, 72, 91, 102
- varianza, 70, 72, 74, 81, 88, 93, 95, 110, 119, 121
- Región de débil singularidad (DS), 48, 82, 129, 130, 215
- Región de máxima singularidad (MS), 48, 82, 129, 130, 136, 215
- Resonadores cargados de forma simple, 264
- Resonadores doblemente cargados, 263
- Respuesta tipo Butterworth, 285, 290
- Respuesta tipo Chebychev, 254, 274, 275, 277
- Resto de la suma de una serie infinita
 - caso bidimensional, 43, 44
 - caso unidimensional, 41–43
- Separaciones entre resonadores, 281
- Separación de acoplo, 267, 269, 281
- Separación entre resonadores, 267, 268, 270, 272, 288
- Serie de imágenes, 33
- Singularidad de las funciones de Green, 24, 50, 103
 - eliminación, 82
 - eliminación mediante la extracción de dos capas, 128, 129, 138, 283
 - eliminación mediante la multiplicación por la distancia (ρ), 83, 87, 138, 249, 268
- Singularidad de los campos electromagnéticos, 30
- Sobre-entrenamiento, 115
- Suma parcial finita
 - caso bidimensional, 44
 - caso unidimensional, 41
- Tensión en la frontera entre dos líneas de transmisión del modelo circuital equivalente, 28, 30
- Teorema de Stone-Weierstrass, 321
- Umbral de las regiones MS y DS, 47
- Voltaje en el generador de intensidad del modelo circuital equivalente, 27, 34, 38
 - parte cuasi-estática, 35–38

Glosario

~

Símbolo para representar las magnitudes definidas en el dominio espectral. Las magnitudes en el dominio espacial no poseen ningún símbolo característico.**30**

(x', y', z')

Coordenadas del punto fuente.**25**

(x, y, z)

Coordenadas del punto observación.**25**

$Ent(gen)$

Valor entero sin signo de la cadena de bits que forman el gen.**243**

$F_\eta(\mathbf{x})$

Salida η -ésima de la RBFNN y de la Red neuronal de Chebychev.**82**

$G_{cA_0}^{xx}, G_{cA_0}^{yy}$

Parte cuasi-estática de la función de Green del potencial magnético vectorial. El método de suma de imágenes espaciales se aplica sobre esta parte de las funciones de Green del medio multicapa apantallado para aquellos puntos observación cercanos a la fuente. Se utiliza en el método híbrido.**34**

$G_{cA_D}^{xx}, G_{cA_D}^{yy}$

Parte dinámica de la función de Green del potencial magnético vectorial se suma directamente en el dominio espectral para aquellos puntos observación cercanos a la fuente. Se utiliza en el método híbrido.**34**

G_{cA}^{xx}

Función de Green del potencial magnético vectorial. En el Capítulo 2 se utiliza el subíndice c que alude a situación de encapsulamiento. Asimismo, en el Capítulo 2 la función de Green del potencial eléctrico escalar del medio multicapa sin apantallar en el dominio espacial se representa como G_A^{xx} . En los capítulos posteriores se prefiere simplemente G_A^{xx} para referirse a la función de Green correspondiente al medio apantallado ya que esta notación es más acorde con los trabajos publicados.**25**

G_{cA}^{yy}

Función de Green del potencial magnético vectorial. En el Capítulo 2 se utiliza el subíndice c que alude a situación de encapsulamiento. Asimismo, en el Capítulo 2 la función de Green del potencial eléctrico escalar del medio multicapa sin apantallar en el dominio espacial se representa como G_A^{yy} . En los capítulos posteriores se prefiere simplemente G_A^{yy} para referirse a la función de Green correspondiente al medio apantallado ya que esta notación es más acorde con los trabajos publicados.**25**

 G_{cV_0}

Parte cuasi-estática de la función de Green del potencial eléctrico escalar. El método de suma de imágenes espaciales se aplica sobre esta parte de las funciones de Green del medio multicapa apantallado para aquellos puntos observación cercanos a la fuente. Se utiliza en el método híbrido.**34**

 G_{cV_D}

Parte dinámica de la función de Green del potencial eléctrico escalar se suma directamente en el dominio espectral para aquellos puntos observación cercanos a la fuente. Se utiliza en el método híbrido.**34**

 G_{cV}

Función de Green del potencial eléctrico escalar. En el Capítulo 2 se utiliza el subíndice c que alude a situación de encapsulamiento. Asimismo, en el Capítulo 2 la función de Green del potencial eléctrico escalar del medio multicapa sin apantallar en el dominio espacial se representa como G_V . En los capítulos posteriores se prefiere simplemente G_V para referirse a la función de Green correspondiente al medio apantallado ya que esta notación es más acorde con los trabajos publicados.**25**

 $I_1(u)$

Intensidad que recorre la línea de transmisión equivalente a la capa (u) en dirección a la capa ($u + 1$) en el modelo circuital.**28**

 $I_2(u)$

Intensidad que recorre la línea de transmisión equivalente a la capa (u) en dirección a la capa ($u - 1$) en el modelo circuital.**28**

 I_g

Valor del generador de intensidad que representa la excitación puntual colocada en z' en el modelo circuital equivalente a la estructura multicapa apantallada.**28**

 $K(,)$

Función de kernel de una SVM. Es la función que define cada neurona.**63**

 L

Anchura de una tira metálica que componen un filtro de líneas acopladas.**101**

L_r

Longitud física de un resonador.**263**

 M

Número de neuronas calculadas por el método de entrenamiento a partir de los N datos de entrenamiento.**65**

 $M(s), E(s)$

Variables auxiliares empleadas en el cálculo de $V_{0,m,n}$. Se utilizan en el método híbrido.**34**

 M_s

Subconjunto del conjunto total de regresores. Se utiliza en la explicación del algoritmo de entrenamiento OLS. De este subconjunto se extraen los M mejores regresores o centros. Generalmente $M_s = N$.**74**

 N

Número de datos de entrenamiento de una red neuronal.**65**

 $N_{a,m,n}^{TE}, N_{a,m,n}^{TM}$

Factores de normalización del potencial magnético vectorial para los modos $TE_{m,n}$ y $TM_{m,n}$ respectivamente.**27**

 $N_{b,m,n}, N_{a,m,n}$

Factores de normalización para que la expansión en funciones modales en el dominio espectral de las funciones de Green forme un conjunto ortonormal. $N_{b,m,n}$ se asigna al potencial eléctrico escalar y $N_{a,m,n}$ al potencial magnético vectorial.**27**

 $N_{b,m,n}^{TE}, N_{b,m,n}^{TM}$

Factores de normalización del potencial eléctrico escalar para los modos $TE_{m,n}$ y $TM_{m,n}$ respectivamente. Se cumple que: $N_{b,m,n}^{TE} = N_{b,m,n}^{TM}$.**27**

 $P_i(\mathbf{x})$

Función neuronal i -ésima de una Red neuronal de Chebychev. Consiste en el producto de n polinomios de Chebychev, donde n es la dimensión del espacio de entrada.**181**

 Q_e

Factor de calidad externo de un filtro. Se utiliza como entrada de una de las RBFNNs que calculan las dimensiones físicas del filtro.**263**

 $R_{M,N}$

Resto de la serie infinita definida en el dominio espectral para el caso bidimensional. Es la serie que se obtiene al restar una suma parcial finita de la serie original. Se emplea en el método de la suma por partes.**41**

R_N

Resto de la serie infinita definida en el dominio espectral para el caso unidimensional. Es la serie que se obtiene al restar una suma parcial finita de la serie original. Se emplea en el método de la suma por partes.**41**

Ratio

Ratio entre el rendimiento del filtro ideal y el rendimiento de un filtro auxiliar. Sirve para calcular el conjunto de parámetros z en la nueva función de rendimiento del AG.**255**

 $Rend(z_{(S_i, \Delta_j)})$

Rendimiento ,generado con la nueva función, de la respuesta S_i en la banda Δ_j en el AG.**255**

 S

Separación entre dos tiras metálicas que componen un filtro de líneas acopladas.**101**

 S_k

Grupo de datos de entrenamiento que sirven para calcular el centro asociado a dicho grupo. Se emplea en el algoritmo de "K-means".**73**

 S_{N-1}

Suma parcial finita de términos tomados de la serie infinita definida en el dominio espectral para el caso unidimensional. Se emplea en el método de la suma por partes.**41**

 S_Q

Separación entre un puerto y el resonador adyacente en los filtros de líneas acopladas.**263**

 S_i

Respuesta i -ésima de un filtro en la que se desea que el cumplimiento de ciertas especificaciones. Se utiliza en el cálculo de los parámetros z de la nueva función de rendimiento del AG.**254**

 S_k

Separación entre dos resonadores.**263**

 $TE_{m,n}$

Modo transversal eléctrico con índices (m, n) .**27**

 $TM_{m,n}$

Modo transversal magnético con índices (m, n) .**27**

 $T_n(x)$

Polinomio de Chebychev de orden n .**173**

 $T_{o(n)}(x)$

Función neuronal de la neurona n -ésima de una Red neuronal de Chebychev cuando el espacio de entrada es unidimensional. Es un polinomio de Chebychev de orden $o(n)$.**177**

$T_{o_j(i)}(x_j)$

Polinomio de orden o_j correspondiente al j -ésimo término del productorio que define la neurona i -ésima cuando el espacio de entrada es multidimensional. Se encuentra asociado a la coordenada j -ésima del espacio de entrada multidimensional. Así, el índice j varía entre 1 y n , donde n es la dimensión del espacio de entrada. El índice o_j varía entre 0 y $p(j) - 1$, donde $p(j)$ es el número de puntos en los que se muestrea la coordenada j -ésima del espacio de entrada multidimensional. Estos puntos son las raíces del polinomio de orden superior $p(j)$.**181**

 $U_{b(S_i, \Delta_j)}$

Umbral de la especificación para la respuesta S_i en la banda Δ_j .**255**

 U_x, U_y

Umbral de las regiones MS y DS. Se halla experimentalmente, generalmente $U_x = U_y$. Se emplea en el método de la suma por partes.**41**

 $V_{0m,n}$

Voltaje de la parte cuasi-estática del modelo circuital equivalente para el modo (m, n) . Se utiliza en el método híbrido.**34**

 $V_1(u)$

Tensión en la frontera existente entre las líneas de transmisión equivalentes a las capas (u) y $(u - 1)$ en el modelo circuital.**28**

 $V_2(u)$

Tensión en la frontera existente entre las líneas de transmisión equivalentes a las capas (u) y $(u + 1)$ en el modelo circuital.**28**

 $V_{m,n}(z, z')$

Voltaje en la coordenada z que produce un generador de intensidad que representa la excitación puntual colocada en z' en el modelo circuital equivalente a la estructura multicapa apantallada.**27**

 W

Anchura de las tiras metálicas que componen un filtro de líneas acopladas.**101**

 W

Anchura de los resonadores en los filtros de líneas acopladas.**263**

 $Z_{c_{m,n}}^{TE}(u), Z_{c_{m,n}}^{TM}(u)$

Impedancia característica de la línea de transmisión correspondiente a la capa (u) para el modo $TE_{m,n}$ y para el modo $TM_{m,n}$ respectivamente en el modelo circuital equivalente.**27**

 $Z_{in}(b)$

Impedancia de entrada de la línea de transmisión equivalente a la capa que se encuentra en contacto con la tapa inferior b en el modelo circuital.**28**

$Z_{in}(t)$

Impedancia de entrada de la línea de transmisión equivalente a la capa que se encuentra en contacto con la tapa superior t en el modelo circuital.28

 $Z_{in}(u)$

Impedancia de entrada de la línea de transmisión equivalente a la capa (u) en el modelo circuital.28

 Δ_j

Banda de frecuencia j -ésima en la que un filtro debe cumplir ciertas especificaciones. Se utiliza en el cálculo de los parámetros \mathbf{z} de la nueva función de rendimiento del AG.254

 $\Psi(m, N)$

Suma parcial (resto) unidimensional en el índice (m). Concretamente es el sumatorio desde el término ($k = 1$) hasta $+\infty$ del producto de $\tilde{G}_{m,N}^{(-1,-k)}$ por $g_{N-1}^{(k+1)}$. Se emplea en el caso bidimensional del método de la suma por partes para convertir el resto $R_{M,N}$ en una serie unidimensional.41

 $\Psi_1(m, N), \Psi_2(m, N)$

Funciones resto unidimensionales similares a $\Psi(m, N)$ utilizadas para evaluar la función de Green del potencial eléctrico escalar. Se emplea en el método de la suma por partes.41

 Σ

Matriz de covarianza de la función gaussiana de una RBFNN. Se emplea para dotar a cada coordenada del espacio de entrada de una varianza propia.65

 $Y(m)$

Serie unidimensional utilizada en la función de Green del potencial eléctrico escalar para aplicar el método de la suma por partes.41

 $Y^{xx}(m)$

Serie unidimensional utilizada en la función de Green del potencial magnético vectorial G_{cA}^{xx} para aplicar el método de la suma por partes.41

 $Y^{yy}(m)$

Serie unidimensional utilizada en la función de Green del potencial magnético vectorial G_{cA}^{yy} para aplicar el método de la suma por partes.41

 α

Factor de ponderación en el cálculo de la varianza gaussiana en el algoritmo de "K-means".73

 α_{jk}

Sirven para ortogonalizar los regresores en el algoritmo OLS. También sirven para calcular los pesos de la capa de salida. El subíndice j se refiere al regresor ortogonal ya seleccionado y el subíndice k a la iteración.74

$\beta_{(i,j)}$

Peso que pondera la importancia de la función de rendimiento de la respuesta S_i en la banda Δ_j en el AG.**254**

 $\beta_{m,n}^u$

Constante de propagación en la línea de transmisión correspondiente a la capa (u) en el modelo circuital equivalente.**27**

 Φ

Matriz de regresores cuando todos los datos de entrenamiento son seleccionados como centros de las neuronas.**65**

 ω

Vector de pesos de la capa de salida de una RBFNN y de una Red neuronal de Chebychev. Si el espacio de salida es multidimensional es una matriz.**65**

 δ_f

Límite de la zona frecuencial definida en torno a una frecuencia de resonancia en la que no se realiza la aproximación de las funciones de Green.**101**

 ϵ_m

Factor de Neumman.**27**

 ϵ_r

Permitividad relativa del dieléctrico que ocupa una capa de la estructura multicapa.**27**

 T

Matriz de regresores en una Red neuronal de Chebychev. Si se muestrea en las raíces de los polinomios de Chebychev adecuados los regresores de esta matriz son ortogonales inicialmente.**177**

 T_n

Regresor n -ésimo en una Red neuronal de Chebychev. En el caso de un espacio de entrada unidimensional es un polinomio de Chebychev de orden $o(n)$ ($T_{o(n)}(x)$) evaluado en las raíces de un polinomio de orden N si el espacio de entrada unidimensional es muestreado en N puntos.**177**

 g_η

Columna η -ésima de la matriz G en el algoritmo OLS.**80**

 $g_{i\eta}^k$

Columna η -ésima de la matriz G en el algoritmo OLS. Es un vector asociado al regresor candidato i -ésimo en la iteración k -ésima que sirve para calcular el ratio de reducción del error.**80**

 $x_{k(z)}$

Vector de entrada de entrenamiento de una Red neuronal de Chebychev cuando el espacio de entrada es multidimensional. Si el espacio de entrada es n -dimensional, entonces cada vector posee n elementos desde $x_{k_1(z)}$ hasta $x_{k_n(z)}$.**181**

μ_r

Permeabilidad del dieléctrico que ocupa una capa de la estructura multicapa.**27**

 ω_i

Peso de la capa de salida de una RBFNN y de una Red neuronal de Chebychev con salida unidimensional.**65**

 $\phi()$

Función de base radial de una RBFNN. Define cada una de las neuronas de una RBFNN.**65**

 ρ

Distancia entre el punto observación y el punto fuente.**82**

 σ

Conductividad del conductor (metal) utilizado en la realización de circuitos impresos.**27**

 σ_f^2

Varianza de la coordenada frecuencia de un dato de entrada en el problema de aproximación de las funciones de Green.**101**

 σ_i^2

Varianza de la función gaussiana de una RBFNN correspondiente a la coordenada i -ésima del espacio de entrada.**65**

 σ_s^2

Varianza de las coordenadas espaciales de un dato de entrada en el problema de aproximación de las funciones de Green.**101**

 $\tan \delta$

Tangente de pérdidas del dieléctrico que ocupa una capa de la estructura multicapa.**27**

 \tanh

Tangente hiperbólica. Se utiliza en la definición de las funciones logísticas de la nueva función de rendimiento del AG.**255**

A

Matriz triangular que contiene los valores α_{ij} en el algoritmo OLS.**74**

C

Matriz de norma ponderada de la función gaussiana de una RBFNN. Se emplea para dotar a cada coordenada del espacio de entrada de una varianza propia.**65**

E

Vector de errores en el algoritmo OLS. Si la salida es multidimensional es una matriz.**74**

G

Matriz resultado del producto de la matriz \mathbf{A} por la matriz de pesos ω en el caso multidimensional. Se emplea en el algoritmo OLS para calcular la reducción del ratio de error.**80**

H

Matriz diagonal resultado del producto $\mathbf{M}^T \mathbf{M}$ en el algoritmo OLS.**74**

M

Matriz de regresores ortogonales. Utilizada en el algoritmo OLS para seleccionar los mejores regresores (centros).**74**

 \mathbf{M}_y

Matriz que almacena los productos escalares de los regresores con los vectores columna de la matriz de valores de salida. Se emplea en la versión mejorada del algoritmo OLS desarrollada en esta tesis.**113**

P

Matriz de regresores de una RBFNN. Es una matriz fruto de la evaluación de los datos de entrenamiento en las neuronas (funciones de base radial con un centro propio). También es la matriz de regresores de una Red neuronal de Chebychev cuando el espacio de entrada es multidimensional.**65**

 \mathbf{P}^+

Matriz pseudo-inversa de la matriz \mathbf{P} . Se utiliza para calcular los pesos de la capa de salida de la RBFNN.**65**

 \mathbf{c}_i

Centro o centroide de la neurona i -ésima perteneciente a una RBFNN.**65**

g

Vector resultado del producto de la matriz \mathbf{A} por el vector de pesos ω en el caso unidimensional. Se emplea en el algoritmo OLS para calcular la reducción del ratio de error.**74**

 \mathbf{m}_i^k

Regresor ortogonalizado del regresor candidato \mathbf{p}_i para formar parte de la base de regresores ortogonales en el algoritmo OLS en la iteración k -ésima.**74**

 \mathbf{m}_k

Regresor ortogonalizado y seleccionado en el algoritmo OLS. Es la columna k -ésima de la matriz \mathbf{M} .**74**

 \mathbf{p}_i

Regresor i -ésimo. Es la columna i -ésima de la matriz \mathbf{P} . Es el fruto de evaluar la función de base radial para un centro determinado en todos los datos de entrenamiento.**74**

x

Vector de datos de entrada de entrenamiento de una red neuronal. En el caso de que el espacio de entrada sea multidimensional es una matriz. En las Redes neuronales de Chebychev para un espacio general n -dimensional $\mathbf{x}_i \in [-1, 1]^n$.**65**

 \mathbf{x}_i

Dato de entrenamiento i -ésimo.**65**

y

Vector de datos de salida de entrenamiento de una red neuronal. En el caso de que el espacio de salida sea multidimensional es una matriz.**65**

 \mathbf{y}_η

Columna η -ésima de la matriz **Y**.**80**

 $\tilde{G}_{m,n}^{(-1,-1)}$

Función de Green en el dominio espectral de índices (m, n) . Se emplea en el método de la suma por partes.**41**

 $\tilde{G}_{m,n}^{(-i,-k)}$

Diferencia entre las funciones $\tilde{G}_{m+1,n}^{(-i+1,-k)}$ y $\tilde{G}_{m,n}^{(-i+1,-k)}$. Es una función recursiva que se emplea en el caso bidimensional del método de la suma por partes.**41**

 $\tilde{G}_n^{(-1)}$

Función de Green en el dominio espectral con índice (n) . Se emplea en el método de la suma por partes.**41**

 $\tilde{G}_n^{(-i)}$

Diferencia entre las funciones $\tilde{G}_{n+1}^{(-i+1)}$ y $\tilde{G}_n^{(-i+1)}$. Es una función recursiva que se emplea en el caso unidimensional del método de la suma por partes.**41**

 φ

Función de activación neuronal del Perceptrón multicapa (MLP).**61**

 $\hat{\mathbf{g}}$

Estima del vector o matriz **g** en el algoritmo OLS.**74**

 ζ

Índice empleado en el cálculo del NMSE.**82**

 a

Longitud de la cavidad en la que se imprimen los circuitos.**101**

 b

Bias o constante de una RBFNN. Es una neurona de valor constante. Favorece la capacidad de aproximación de una RBFNN. Suele tomar un valor igual a la unidad de forma que es el peso o pesos correspondientes a esta neurona los que añaden la constante a la salida.**65**

b

Longitud de la cavidad en la que se imprimen los circuitos.**101**

 c_j

Coeficiente j -ésimo de la fórmula de aproximación de Chebychev.**173**

 d_1

Límite entre la primera zona observación y las zonas intermedias en el algoritmo de división del espacio de entrada.**101**

 $d_{\omega}^{TE}, d_{\omega}^{TM}$

Variables auxiliares empleadas en el cálculo de $V_{0_{m,n}}$. Se utilizan en el método híbrido.**34**

 err_i^k

Ratio de reducción del error del regresor candidato i -ésimo en la iteración k -ésimo en el algoritmo OLS. Mide la contribución del regresor candidato a la energía de salida.**74**

 $f(k_{x_m}, x, x')$

Función sinusoidal dependiente de las abscisas del punto observación y fuente usada en la definición de las funciones de Green del medio apantallado.**26**

 $f_n^{(+1)}$

Función sinusoidal f_n . Se emplea en el método de la suma por partes.**41**

 $f_n^{(+2)}$

Sumatorio la función sinusoidal $f_n^{(+1)}$ desde el término $(n + 1)$ hasta $+\infty$. Se emplea en el método de la suma por partes.**41**

 $f_n^{(+i)}$

Sumatorio la función sinusoidal $f_n^{(+i-1)}$ desde el término $(n + 1)$ hasta $+\infty$. Se emplea en el método de la suma por partes.**41**

 f_o

Frecuencia de resonancia de los resonadores que forman el filtro. Se utiliza como entrada de una de las RBFNNs que calculan las dimensiones físicas del filtro.**263**

 f_{p_1}, f_{p_2}

Par de frecuencias que permiten hallar el coeficiente de acoplo entre dos resonadores.**263**

 $g(k_{y_n}, y, y')$

Función sinusoidal dependiente de las ordenadas del punto observación y fuente usada en la definición de las funciones de Green del medio apantallado.**26**

 g_i^k

Valor del vector \mathbf{g} para el caso unidimensional en el algoritmo OLS. Está asociado al candidato i -ésimo en la iteración k -ésima y sirve para calcular el ratio de reducción del error.**74**

$g_n^{(+1)}$

Función sinusoidal g_n . Se emplea en el método de la suma por partes.**41**

 $g_n^{(+2)}$

Sumatorio la función sinusoidal $g_n^{(+1)}$ desde el término $(n + 1)$ hasta $+\infty$. Se emplea en el método de la suma por partes.**41**

 $g_n^{(+k)}$

Sumatorio la función sinusoidal $g_n^{(+k-1)}$ desde el término $(n + 1)$ hasta $+\infty$. Se emplea en el método de la suma por partes.**41**

 h

Altura de una capa de dieléctrico de la estructura multicapa.**101**

 k

Coefficiente de acoplo entre resonadores. Se utiliza como entrada de una de las RBFNNs que calculan las dimensiones físicas del filtro.**263**

 k

Índice de la capa de influencia en la serie de imágenes. Se utiliza en el método híbrido.**32**

 $k_{\rho_{m,n}}$

Variable radial compleja definida en el dominio espectral. Vector de propagación transversal.**27**

 k_m

Índice de la raíz del polinomio de orden superior correspondiente a la coordenada m -ésima. Si el polinomio de orden superior es de orden $p(m)$, entonces este índice varía entre 1 y $p(m)$.**181**

 k_{x_m}

Número de onda transversal en la abscisa x .**25**

 k_{y_n}

Número de onda transversal en la ordenada y .**25**

 $o(n)$

Orden del polinomio de Chebychev de la neurona n -ésima de una Red neuronal de Chebychev. Si el espacio de entrada unidimensional es muestreado en las N raíces del polinomio de orden superior N , entonces $o(n)$ puede tomar un valor entre 0 y $N - 1$.**177**

 $o_j(i)$

Orden del polinomio asociado a la coordenada j -ésima de la neurona i -ésima en una Red neuronal de Chebychev cuando el espacio de entrada es multidimensional.**177**

 p

Superíndice que indica el tipo de modo TE o TM . Por ejemplo, se emplea en la impedancia característica del modelo circuital equivalente $Z_{c_{m,n}}^p(i)$ **27**

$p(m)$

Número de puntos de muestreo de la coordenada m -ésima del espacio de entrada cuando éste es multidimensional. Es también el orden del polinomio de orden superior de esta coordenada.**181**

 q

Dimensión del espacio de salida en el caso multidimensional.**80**

 s_x, s_y

Funciones signo empleadas para definir las funciones de Green apantalladas en el método híbrido.**32**

 t

Distancia de "tapping". Distancia entre un puerto y la base del resonador adyacente en los filtros de tipo horquilla ("hairpin")**263**

 u

Capa de la estructura multicapa apantallada y línea de transmisión del modelo circuital equivalente.**27**

 x_{inf}, x_{sup}

Límites inferior y superior de la función objetivo. Se utiliza en los problemas de aproximación mediante las Redes neuronales de Chebychev.**173**

 $x_{k_m}(z)$

Raíz del polinomio de orden superior $p(m)$ correspondiente a la coordenada m -ésima. Se utiliza para formar el conjunto de entrenamiento en las Redes neuronales de Chebychev cuando el espacio de entrada es multidimensional.**181**

 x_k

Raíz de un polinomio de Chebychev. Si el orden del polinomio es n , entonces el índice k varía desde 1 hasta n . Se utiliza para formar el conjunto de entrenamiento en las Redes neuronales de Chebychev cuando el espacio de entrada es unidimensional.**173**

 x_{max}, x_{min}

Incremento máximo y mínimo que utiliza para definir el valor de cada gen del cromosoma en el AG. El incremento se aplica sobre un determinado valor que se fija antes del inicio del AG.**243**

 x_m

Extremo m -ésimo del polinomio de Chebychev de orden n . El índice n varía desde 0 hasta $n + 1$.**173**

 x_{or}

Variable independiente de la función objetivo definida en el rango original. Se utiliza en los problemas de aproximación mediante las Redes neuronales de Chebychev.**173**

z

Parámetro que controla la pendiente de las funciones logísticas en la nueva función de rendimiento del AG. Mediante este parámetro se controla el grado de gratificación por el cumplimiento de una especificación o la penalización por un incumplimiento. **243**

z

Índice de los datos de entrenamiento de una Red neuronal de Chebychev. Varía entre 0 y N. **181**

 $z(s_i, \Delta_j)$

Parámetro z de la respuesta S_i en la banda Δ_j en la nueva función de rendimiento del AG. **254**

"Golden section search"

Técnica de optimización que se basa en el examen de los valores de la función objetivo en diversos puntos. No utiliza el gradiente de la función objetivo. **263**

"K-means" o K-medias: algoritmo de agrupación en K-medias.

Método no supervisado de entrenamiento de una RBFNN. **73**

Algoritmo de optimización minimax

Algoritmo de optimización que busca minimizar la máxima diferencia encontrada entre el valor actual y el valor deseado. Utiliza el gradiente de la función objetivo. **263**

Algoritmo genético (AG)

Técnica de optimización global basada en la mejora de las características de una población de individuos. Aplica mecanismos por los cuales las especies mejoran su capacidad de adaptación al medio. **243**

Capa de influencia

Agrupación de conjuntos de imágenes básicos. Dependiendo de la cercanía a la fuente cada capa de influencia constará de una cantidad determinada de conjuntos de imágenes básicos. Por ejemplo, la primera capa de influencia está formada por un conjunto imágenes básico que contiene la carga original y las tres cargas imágenes, la segunda capa de influencia consta de ocho conjuntos de imágenes básicos alrededor del anterior y así sucesivamente. Se emplea en el método híbrido. **32**

Conjunto de imágenes básico

Conjunto de cuatro imágenes que permite la eliminación de las paredes metálicas en el problema de la evaluación de las funciones de Green. Así, permiten la utilización de las funciones de Green de espacio libre asociadas. Para representar completamente el problema se deben usar infinitos grupos de imágenes básicos. Se emplea en el método híbrido. **32**

Convergencia prematura

Problema que puede surgir en el AG. Consiste en la agrupación de los individuos de la población en torno a un individuo que sobresale respecto al resto. Se evita mediante la utilización de una técnica de medición del rendimiento adecuada como la desarrollada en este trabajo. **243**

Cromosoma

Conjunto de genes que determinan las características del individuo en el AG.243

Elitismo

Mecanismo que en el AG permite la conservación de algunos de los mejores individuos en la generación de una nueva población de individuos.243

FOLS: "Fast orthogonal least squares"

Versión rápida del algoritmo OLS.113

Gen

Conjunto de bits que codifican una característica de un individuo del AG. En este trabajo cada gen codifica una dimensión física de un filtro.243

Individuo

Unidad sobre la que actúa el AG. En este trabajo cada individuo es un filtro con dimensiones propias pero del mismo orden que el resto de individuos.243

Mutación

Cambio en el valor de un bit en un gen de un individuo en el AG.243

Método de la suma por partes

Método empleado para evaluar las funciones de Green en la región DS (puntos observación alejados de la fuente. Se aplica sobre la serie modal definida en el dominio espectral. Calcula el sumatorio final mediante diferencias entre funciones de lenta variación y sumatorios analíticos de funciones sinusoidales.40

Método híbrido

Método utilizado para calcular las funciones de Green de los potenciales en la región MS. Este método emplea la suma de imágenes para calcular la parte estática de las funciones de Green mientras que suma directamente la parte dinámica en el dominio espectral.30

NMSE: Error cuadrático medio normalizado

Es el tipo de error empleado para medir la calidad de la aproximación de las RBFNNs.82

OLS: Mínimos cuadrados ortogonales ("Orthogonal least squares")

Método de entrenamiento supervisado de una RBFNN.74

Población

Conjunto de todos los individuos en el AG.243

Punto de cruce

Punto en el que se divide la información genética en un determinado gen en dos individuos seleccionados como padres en el AG. La información genética es intercambiada en este punto para producir los genes de la descendencia.²⁴³

MLP: Perceptrón multicapa.

Es un tipo de red neuronal. Su estructura puede constar de varias capas ocultas de neuronas. En este trabajo se han empleado para aproximar las funciones de Green de los potenciales con el objetivo de establecer una comparación con las RBFNNs.⁶¹

RBFNN: Redes neuronales de funciones de base radial ("Radial basis functions neural networks").

Es un tipo de red neuronal, utilizada en esta tesis para aproximar las funciones de Green. Posee tres capas: la capa de entrada, la capa oculta o intermedia y la capa de salida. La capa de oculta está formada por neuronas definidas por una función de base radial común. Esta capa realiza una transformación no lineal sobre los datos de entrada. La capa de salida está compuesta por una serie de pesos que llevan a cabo una transformación lineal.⁶⁵

SVM: Máquinas de vectores soporte.

Es un tipo de red neuronal. Su estructura es similar a la de las RBFNNs. En este trabajo se han empleado para aproximar las funciones de Green de los potenciales con el objetivo de establecer una comparación con las RBFNNs.⁶³

Redes neuronales de Chebychev

Es un tipo de red neuronal, utilizada en esta tesis para aproximar las funciones de Green. Su estructura es similar a la de una RBFNN. Posee tres capas: la capa de entrada, la capa oculta o intermedia y la capa de salida. La capa de oculta está formada por neuronas definidas por un polinomio de Chebychev en el caso unidimensional o por un producto de polinomios de Chebychev en el caso multidimensional. Esta capa realiza una transformación no lineal sobre los datos de entrada. La capa de salida está compuesta por una serie de pesos que llevan a cabo una transformación lineal.¹⁷⁷

Rendimiento

Rendimiento de un individuo de la población en el AG. En este trabajo indica el grado de cumplimiento de las especificaciones por parte de un filtro.²⁴³

Selección de centros aleatoria.

Método no supervisado de entrenamiento de una RBFNN.⁷²

SMO: Algoritmo de optimización secuencial

Método de entrenamiento de una SVM.⁶³

SVM-Light

Método de entrenamiento de una SVM.⁶³

Técnica de la ruleta

Técnica de selección de un individuo como padre en el AG.243

VC: Dimensión de Vapnik-Chervonenkis

Dimensión que debe ser minimizada para construir una SVM.63

Bibliografía

- [Ackley et al., 1985] Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169.
- [Alos y Guglielmi, 1997] Alos, J. T. and Guglielmi, M. (1997). Simple and effective EM-based optimization procedure for microwave filters. *IEEE Transactions on Microwave Theory and Techniques*, 45(6):856–858.
- [Bai y Zhang, 2002] Bai, Y. and Zhang, L. (2002). Genetic algorithm based self-growing training for rbf neural networks. *Proceeding of the 2002 international joint conference on neural networks*, 1-3:840–845.
- [Bandler et al., 1994a] Bandler, J., Biernacki, R., Chen, S. H., D.G. Swanson, J., and Ye, S. (1994a). Microstrip filter design using direct em field simulation. *IEEE Transactions on Microwave Theory and Techniques*, 42(7):1353–1359.
- [Bandler et al., 1994b] Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny, P. A., and Hemmers, R. H. (1994b). Space mapping technique for electromagnetic simulation. *IEEE Transactions on Microwave Theory and Techniques*, 42(12):2536–2544.
- [Bandler et al., 1985] Bandler, J. W., Kellerman, W., and Madsen, K. (1985). A superlinearly convergent minimax algorithm for microwave circuit design. *IEEE Transactions on Microwave Theory and Techniques*, 33(12):1519–1530.
- [Bandler et al., 1993] Bandler, J. W., Ye, S., Biernacki, R. M., Chen, S. H., and Swansom, D. G. (12-14 june, 1993). Minimax microstrip filter design using direct em field simulation. *Microwave Symposium Digest, 1993., IEEE MTT-S International*, 12:889–892.
- [Barret, 1955] Barret, R. M. (1955). Microwave printed circuits - a historical survey. *IEEE Transactions on Microwave Theory and Techniques*, 3:1–9.
- [Borji y Safavi-Naeini, 2004] Borji, A. and Safavi-Naeini, S. (2004). Rapid calculation of the green's function in a rectangular enclosure with application to conductor loaded cavity resonators. *IEEE Transactions on Microwave Theory and Techniques*, 52(7):1724–1731.
- [Boser et al., 1992] Boser, B., Guyon, I., and Vapnik, V.Ñ. (1992). A training algorithm for optimal margin classifiers. *Fifth annual workshop on computational learning theory*, pages 144–152.

- [Broomhead y Lowe, 1988] Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.
- [Chapman, 1998] Chapman, S. J. (1998). *Fortran 90/95 for Scientists and engineers*. MacGraw-Hill.
- [Chen y Tseng, 2004] Chen, C.-S. and Tseng, C.-S. (2004). Performance comparison between the training method and the numerical method of the orthogonal neural network in function approximation. *International journal of intelligent systems*, 19(12):1257–1275.
- [Chen, 1988] Chen, J. B. S. (1988). Circuit optimization: the state of the art. *IEEE Transactions on Microwave Theory and Techniques*, 36(2):424–443.
- [Chen et al., 1991] Chen, S., Cowan, C. F.Ñ., and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309.
- [Chen et al., 1992] Chen, S., Grant, P. M., and Cowan, C. F.Ñ. (1992). Orthogonal least squares algorithm for training multi-output radial basis function networks. *IEE Proceedings-F radar and Signal Processing*, 139(6):378–384.
- [Chen y Wigger, 1995] Chen, S. and Wigger, J. (1995). Fast orthogonal least squares algorithm for efficient subset model selection. *IEEE Transactions on Signal Processing*, 43(7):1713–1715.
- [Chng et al., 1995] Chng, E. S., Chen, S., and Mulgrew, B. (1995). Efficient computational schemes for the orthogonal least squares algorithm. *IEEE Transactions on Signal Processing*, 43(1):373–376.
- [Christodoulou y Georgipoulos, 2001] Christodoulou, C. and Georgipoulos, M. (2001). *Applications of Neural Networks in Electromagnetics*. Artech House.
- [Cohn, 1957] Cohn, S. B. (1957). Direct-coupled-resonator filters. *Proceedings of the IRE*, 45(2):187–196.
- [Cohn, 1958] Cohn, S. B. (1958). Parallel-coupled transmission-line-resonator filters. *IRE Transactions of Microwave Theory and Techniques*, 6(2):223–231.
- [Collin, 1960] Collin, R. E. (1960). *Field Theory of Guided Waves*. McGraw-Hill, New York, USA.
- [Collin, 1966] Collin, R. E. (1966). *Foundations for Microwave Engineering*. McGraw-Hill.
- [Collobert y Bengio, 2001] Collobert, R. and Bengio, S. (2001). Svmtorch: Support vector machines for large-scale regression problems. *Journal of machine learning Research*, 1(2):143–160.
- [Cortes y Vapnik, 1995] Cortes, C. and Vapnik, V.Ñ. (1995). Support vector networks. *Machine Learning*, 20:273–297.
- [Cover, 1965] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334.
- [Creech et al., 1997] Creech, G. L., Paul, B. J., Lesniak, C., Jenkins, T. J., and Calcaterra, M. C. (1997). Artificial neural networks for fast and accurate em-cad of microwave circuits. *IEEE Transactions on Microwave Theory and Techniques*, 45(5):794–802.

- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. control, signals, systems*, 2:303–314.
- [der Malsburg, 1973] der Malsburg, C. V. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kibernetik*, 14:85–100.
- [Devabhaktuni et al., 2001] Devabhaktuni, V., Yagoub, M., and Zhang, Q. (2001). A robust algorithm for automatic development of neural-network models for microwave applications. *IEEE Transactions on Microwave Theory and Techniques*, 49(12):2282–2291.
- [Dorigo et al., 2006] Dorigo, M., Birattari, M., and Stützle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39.
- [Dunleavy y Katehi, 1988a] Dunleavy, L. P. and Katehi, P. B. (1988a). A generalized method for analyzing shielded thin microstrip discontinuities. *IEEE Transactions on Microwave Theory and Techniques*, 36(12):1758–1766.
- [Dunleavy y Katehi, 1988b] Dunleavy, L. P. and Katehi, P. B. (1988b). Shielding effects in microstrip discontinuities. *IEEE Transactions on Microwave Theory and Techniques*, 36(12):1767–1774.
- [Eisenhart y Khan, 1971] Eisenhart, R. L. and Khan, P. J. (1971). Theoretical and experimental analysis of a waveguide mounting structure. *IEEE Transactions on Microwave Theory and Techniques*, 19:706–719.
- [Eleftheriades et al., 1995] Eleftheriades, G., Mosig, J., and Guglielmi, M. (1995). An efficient mixed potential integral equation technique for the analysis of shielded mmic's. *Proc. 25th European Microwave Conference*, pages 825–829.
- [Eleftheriades et al., 1996] Eleftheriades, G. V., Mosig, J. R., and Guglielmi, M. (1996). A fast integral equation technique for shielded planar circuits defined on nonuniform meshes. *IEEE Transactions on Microwave Theory and Techniques*, 44(12):2293–2296.
- [Fedi et al., 1999] Fedi, G., Gaggelli, A., Manetti, S., and Pelosi, G. (1999). Direct-coupled cavity filters design using a hybrid feedforward neural network finite elements procedure. *International Journal of RF and Microwave CAE*, 9(3):287–296.
- [Fernandez-Redondo et al., 2004] Fernandez-Redondo, M., Hernandez-Espinosa, C., Ortiz-Gomez, M., and Torres-Sospedra, J. (2004). Gradient descent training of radial basis functions. *Advances in neural networks*, 3173:229–234.
- [Flake y Lawrence, 2002] Flake, G. W. and Lawrence, S. (2002). Efficient svm regression training with smo. *Machine Learning*, 46(1-3):271–290.
- [Garcia et al., 2004a] Garcia, J. P., Pereira, F. D. Q., Rebenague, D. C., and Melcon, A. A. (2004a). Using neural networks for the calculation of the multilayered shielded green's functions. *Proceedings of Progress in Electromagnetics Research Symposium, PIERS*.
- [Garcia et al., 2004b] Garcia, J. P., Pereira, F. D. Q., Rebenague, D. C., Tornero, J. L. G., and Melcon, A. A. (2004b). A neural network method for the multilayered shielded Green's function calculation. *The 34th european microwave conference*.

- [Garcia et al., 2005a] Garcia, J. P., Pereira, F. D. Q., Rebenaque, D. C., Tornero, J. L. G., and Melcon, A. A. (2005a). A neural network method for the analysis of multilayered shielded microwave circuits. *IEEE International Microwave Symposium Digest*.
- [Garcia et al., 2006a] Garcia, J. P., Pereira, F. D. Q., Rebenaque, D. C., Tornero, J. L. G., and Melcon, A. A. (2006a). A multilayered shielded microwave circuit design method based on genetic algorithms and neural networks. *IEEE International Microwave Symposium Digest*.
- [Garcia et al., 2006b] Garcia, J. P., Pereira, F. D. Q., Rebenaque, D. C., Tornero, J. L. G., and Melcon, A. A. (2006b). A neural-network method for the analysis of multilayered shielded microwave circuits. *IEEE Transactions on Microwave Theory and Techniques*, 54(1):309–320.
- [Garcia et al., 2006c] Garcia, J. P., Pereira, F. D. Q., Rebenaque, D. C., Tornero, J. L. G., and Melcon, A. A. (2006c). Un método para el diseño de circuitos de microondas apantallados basado en redes neuronales y algoritmos genéticos. *XXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*.
- [Garcia et al., 2009] Garcia, J. P., Pereira, F. D. Q., Rebenaque, D. C., Tornero, J. L. G., and Melcon, A. A. (2009). A new neural network technique for the design of multilayered microwave shielded bandpass filters. *International Journal of RF and Microwave Computer-Aided Engineering*.
- [Garcia et al., 2004c] Garcia, J. P., Rebenaque, D. C., Pereira, F. D. Q., Tornero, J. L. G., and Melcon, A. A. (2004c). Fast and efficient calculation of the multilayered shielded Green's functions employing neural networks. *Microwave and Optical Technology Letters*, 44(1):61–66.
- [Garcia et al., 2004d] Garcia, J. P., Rebenaque, D. C., Pereira, F. D. Q., Tornero, J. L. G., and Melcon, A. A. (2004d). Redes neuronales aplicadas al cálculo de las funciones de green en un medio multicapa encapsulado. *XIX Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*.
- [Garcia et al., 2005b] Garcia, J. P., Rebenaque, D. C., Pereira, F. D. Q., Tornero, J. L. G., and Melcon, A. A. (2005b). Un método neuronal para el análisis de circuitos de microondas multicapa y apantallados. *XX Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*.
- [Garcia et al., 2007] Garcia, J. P., Rebenaque, D. C., Pereira, F. D. Q., Tornero, J. L. G., and Melcon, A. A. (2007). Una nueva técnica de diseño de filtros de microondas multicapa apantallados basada en redes neuronales. *XXII Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*.
- [Gentili et al., 1997] Gentili, G. G., Garcia-Castillo, L. E., Salazar-Palma, M., and Perez-Martinez, F. (1997). Green's function analysis of single and stacked rectangular microstrip patch antennas enclosed in a cavity. *IEEE Transactions on Antennas and Propagation*, 45(4):573–579.
- [Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization and machine learning*. Addison Wesley Longman.
- [Golub y Loan, 1996] Golub, G. H. and Loan, C. G. V. (1996). *Matrix computations*. Johns Hopkins University Press.
- [Gradshteyn y Ryzhik, 1965] Gradshteyn and Ryzhik (1965). *Table of Integrals, Series, and Products*. Academic Press Inc., New York, 1965.

- [Grossberg, 1980] Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, 87:1–51.
- [Guglielmi, 1994] Guglielmi, M. (1994). Simple cad procedure for microwave filters and multiplexers. *IEEE Transactions on Microwave Theory and Techniques*, 42(7):1347–1352.
- [Harrington, 1987] Harrington, R. F. (1987). Matrix methods for field problems. *IEEE Transactions on Microwave Theory and Techniques*, 35:726–730.
- [Harrington, 1993] Harrington, R. F. (1993). *Field computation by moment methods*. IEEE Press.
- [Hashemi-Yeganeh, 1995] Hashemi-Yeganeh, S. (1995). On the summation of double infinite series field computations inside rectangular cavities. *IEEE Transactions on Microwave Theory and Techniques*, 43(3):641–646.
- [Haykin, 1999] Haykin, S. (1999). *Neural Networks, a comprehensive foundation*. Prentice Hall International.
- [Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.
- [Hill y Tripathi, 1991] Hill, A. and Tripathi, V. K. (1991). An efficient algorithm for the three-dimensional analysis of passive microstrip components and discontinuities for microwave and millimeter-wave integrated circuits. *IEEE Transactions on Microwave Theory and Techniques*, 39(1):83–91.
- [Hong y Lancaster, 2001] Hong, J.-S. and Lancaster, M. J. (2001). *Microstrip filters for RF/Microwave applications*. John Wiley and Sons.
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79:2554–2558.
- [Horng et al., 1993] Horng, T., Wang, C., and Alexopoulos, N. G. (1993). Microstrip circuit design using neural networks. *IEEE International Microwave Symposium Digest*, 1:413–416.
- [Hornik et al., 1989] Hornik, K. M., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.
- [Hwang et al., 1994] Hwang, J.-N., Lay, S.-R., Maechler, M., Douglas, R., and Schimert, J. (1994). Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks*, 5(3):342–353.
- [Joachims, 1999] Joachims, T. (1999). *Making large-Scale SVM Learning Practical en Advances in Kernel Methods - Support Vector Learning - B. Schölkopf and C. Burges and A. Smola (ed.)*. MIT Press.
- [Joachims, 2002] Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms de la serie The Springer International Series in Engineering and Computer Science*. Springer.

- [Kabir et al., 2008] Kabir, H., Wang, Y., Yu, M., and Zhang, Q.-J. (2008). Neural network inverse modeling and applications to microwave filter design. *IEEE Transactions on Microwave Theory and Techniques*, 56(4):867–879.
- [Kampolis et al., 2004] Kampolis, L., Karangelos, E., and Giannakoglou, K. (2004). Gradient-assisted radial basis function networks: theory and applications. *Applied mathematical modelling*, 28(2):197–209.
- [Kennedy y Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 4:1942–1948.
- [Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.
- [Kuo y Cheng, 2004] Kuo, J.-T. and Cheng, H.-S. (2004). Design of quasi-elliptic function filters with a dual-passband response. *IEEE Microwave and Wireless Components Letters*, 14(10):472–474.
- [Kwok y Yeung, 1997] Kwok, T.-Y. and Yeung, D.-Y. (1997). Objective functions for training new hidden units in constructive neural networks. *IEEE Transactions on Neural Networks*, 8(5):1131–1148.
- [Lee y Jeng, 1998] Lee, T.-T. and Jeng, J.-T. (1998). The chebyshev-polynomials-based unified model neural networks for function approximation. *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, 28(6):925–935.
- [Lippman, 1989] Lippman, R. P. (1989). Pattern classification using neural networks. *IEEE Communications Magazine*, 27(11):47–50, 59–64.
- [Álvarez Melcón, 1998] Álvarez Melcón, A. (1998). *Applications of the integral equation technique to the analysis and synthesis of multilayered printed shielded microwave circuits and cavity backed antennas*. Tesis Doctoral, EPFL, Lausanne.
- [Ma y Khorasani, 2005] Ma, L. and Khorasani, K. (2005). Constructive feedforward neural networks using hermite polynomial activation functions. *IEEE Transactions on Neural Networks*, 16(4):821–833.
- [Marcuvitz, 1951] Marcuvitz, N. (1951). *Waveguide Handbook*. McGraw-Hill.
- [Matthaei et al., 1980] Matthaei, G. L., Young, L., and Jones, E. M. T. (1980). *Microwave filters, impedance-matching networks, and coupling structures*. Artech House.
- [McCulloch y Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.
- [Melcon y Mosig, 2000] Melcon, A. A. and Mosig, J. R. (2000). Two techniques for the efficient numerical calculation of the Green’s functions for planar shielded circuits and antennas. *IEEE Transactions on Microwave Theory and Techniques*, 48(9):1492–1504.
- [Melcon et al., 1999] Melcon, A. A., Mosig, J. R., and Guglielmi, M. (1999). Efficient CAD of boxed microwave circuits based on arbitrary rectangular elements. *IEEE Transactions on Microwave Theory and Techniques*, 47(7):1045–1058.

- [Mendel, 1995] Mendel, J. M. (1995). Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377.
- [Mercer, 1909] Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Transactions of the London Philosophical Society (A)*, 209:415–446.
- [Micchelli, 1986] Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive approximation*, 2:11–12.
- [Michalski y Mosig, 1997] Michalski, K. A. and Mosig, J. R. (1997). Multilayered media Green’s functions in integral equation formulations. *IEEE Transactions on Antennas and Propagation*, 45(3):508–519.
- [Minsky y Papert, 1956] Minsky, M. L. and Papert, S. A. (1956). *Perceptrons*. Cambridge, MA, MIT Press.
- [Mishra y Patnaik, 1999] Mishra, R. K. and Patnaik, A. (1999). Neurospectral computation for complex resonant frequency of microstrip resonators. *IEEE Microwave and Guided Wave Letters*, 9(9):351–453.
- [Mishra y Patnaik, 2003] Mishra, R. K. and Patnaik, A. (2003). Designing rectangular patch antenna using the neurospectral method. *IEEE Transactions on Antennas and Propagation*, 51(8):1914–1921.
- [Mosig, 1989] Mosig, J. R. (1989). Integral-equation techniques for three-dimensional microstrip structures. *Chapter 6 in: Review of Radio Science. Edited by Ross Stone, URSI-Oxford Science Publications*, pages 127–152.
- [Namatame y Ueda, 1992] Namatame, A. and Ueda, N. (1992). Pattern classification with chebyshev neural networks. *International Journal of Neural Networks*, 3:23–31.
- [Niyogi y Girosi, 1996] Niyogi, P. and Girosi, F. (1996). On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural computation*, 8:819–842.
- [Nocedal y Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer-Verlag.
- [Osowski y Markiewicz, 2005] Osowski, S. and Markiewicz, T. (2005). Ols versus svm approach to learning of rbf networks. *Proceedings of International Joint Conference on Neural Networks*, 2:1051–1056.
- [Pan y Wolff, 1994a] Pan, S.-G. and Wolff, I. (1994a). Scalarization of dyadic spectral Green’s functions and network formalism for three-dimensional full-wave analysis of planar lines and antennas. *IEEE Transactions on Microwave Theory and Techniques*, 42(11):2118–2127.
- [Pan y Wolff, 1994b] Pan, S.-G. and Wolff, I. (1994b). Scalarization of dyadic spectral green’s functions and network formalism for three-dimensional full-wave analysis of planar lines and antennas. *IEEE Transactions on Microwave Theory and Techniques*, 42(11):2118–2127.
- [Park y Sandberg, 1991] Park, J. and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural computation*, 3:246–257.

- [Park y Nam, 1997] Park, M.-J. and Nam, S. (1997). Rapid calculation of the Green's function in the shielded planar structures. *IEEE Microwave and Guided Wave Letters*, 7(10):326–328.
- [Park y Nam, 1998] Park, M.-J. and Nam, S. (1998). Rapid summation of the Green's function for the rectangular waveguide. *IEEE Transactions on Microwave Theory and Techniques*, 46(12):2164–2166.
- [Phillips, 2003] Phillips, G. M. (2003). *Interpolation and approximation by polynomials*. Springer-Verlag, New York.
- [Poggio y Girosi, 1990] Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497.
- [Powell, 1988] Powell, M. J. D. (1988). Radial basis function approximations to polynomials. *Proceedings of Numerical Analysis*, pages 223–241.
- [Pozar, 1998] Pozar, D. M. (1998). *Microwave Engineering 2nd Ed.* John Wiley and Sons.
- [Press et al., 1997] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1997). *Numerical recipes in Fortran 77, Second edition*. Cambridge University Press.
- [Purwar et al., 2007] Purwar, S., Kar, I.Ñ., and Jha, A.Ñ. (2007). On-line system identification of complex systems using chebyshev neural networks. *Applied soft computing*, 7(1):364–372.
- [Rahmat-Samii y Michielsen, 1999] Rahmat-Samii, Y. and Michielsen, E. (1999). *Electromagnetic optimization by genetic algorithms*. Wiley Series in microwave and optical engineering, John Wiley and Sons.
- [Railton y Meade, 1992] Railton, C. J. and Meade, S. A. (1992). Fast rigorous analysis of shielded planar filters. *IEEE Transactions on Microwave Theory and Techniques*, 40(5):978–985.
- [Rautio, 1986] Rautio, J. C. (1986). *A time-harmonic electromagnetic analysis of shielded microstrip circuits*. Ph.D. thesis Syracuse University.
- [Rautio y Harrington, 1987] Rautio, J. C. and Harrington, R. F. (1987). An electromagnetic time-harmonic analysis of shielded microstrip structures. *IEEE Transactions on Microwave Theory and Techniques*, 35(8):726–730.
- [Rochester et al., 1956] Rochester, N., Holland, J. H., Haibt, L. H., and Duda, W. L. (1956). Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on Information Theory*, IT-2:80–93.
- [Rosenblatt, 1956] Rosenblatt, F. (1956). The perceptron: a probabilistic model of information storage and organization in the brain. *Annals of Mathematical Statistics*, 27:832–837.
- [Rosenblatt, 1960] Rosenblatt, F. (1960). On the convergence of reinforcement procedures in simple perceptrons. *Cornell Aeronautical Laboratory Report*.
- [Rudin, 1976] Rudin, W. (1976). *Principles of mathematical analysis*. McGraw-Hill: Tercera edición.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations of back-propagation errors. *Nature (London)*, 323:533–536.

- [Schwenker et al., 2001] Schwenker, F., Kestler, H. A., and Palm, G. (2001). Three learning phases for radial-basis-function networks. *Neural Networks*, 14(4-5):439–458.
- [Selleri et al., 2002] Selleri, S., Manetti, S., and Pelosi, G. (2002). Neural network applications in microwave device design. *International Journal RF and Microwave CAE*, 12:90–97.
- [Sherstinsky y Picard, 1996] Sherstinsky, A. and Picard, R. W. (1996). On the efficiency of the orthogonal least squares training method for radial basis function networks. *IEEE Transactions on Neural Networks*, 7(1):195–200.
- [Silvester, 1969] Silvester, P. (1969). A general high-order finite-element waveguide analysis program. *IEEE Transactions on Microwave Theory and Techniques*, 17:204–210.
- [Soliman et al., 2004] Soliman, E. A., Bakr, M. H., and Nikolova, N. K. (2004). Neural networks-method of moments (nn-mom) for the efficient filling of the coupling matrix. *IEEE Transactions on Antennas and Propagation*, 52(6):1521–1529.
- [Soliman et al., 2003] Soliman, E. A., El-Gamal, M. A., and Abdelmageed, A. K. (2003). Neural network model for the efficient calculation of Green’s functions in layered media. *International Journal of RF and Microwave CAE*, 13(2):128–135.
- [Su y Kuo, 2005] Su, K.-Y. and Kuo, J.-T. (2005). Application of two dimension nonuniform fast fourier transform (2-d nufft) technique to analysis of shielded microstrip circuits. *IEEE Transactions on Microwave Theory and Techniques*, 53(3):993–999.
- [Tseng y Yang, 1995] Tseng, C.-S. and Yang, S.-S. (1995). A new orthogonal neural network. *IEEE International Conference on Neural Networks*, 1(27):296–299.
- [Ureel y Zutter, 1997] Ureel, J. and Zutter, D. D. (1997). Gradient-based minimax optimization of planar microstrip structures with the use of electromagnetic simulations. *International Journal of RF and Microwave CAE*, 7(1):29–36.
- [Uttley, 1956] Uttley, A. M. (1956). A theory of the mechanism of learning based on the computation of conditional probabilities. *Proceedings of the First International Conference on Cybernetics*.
- [Vai y Prasad, 1995] Vai, M. and Prasad, S. (1995). Microwave circuit analysis and design by a massively distributed computing network. *IEEE Transactions on Microwave Theory and Techniques*, 43:1087–1094.
- [Vai et al., 1998] Vai, M. M., Wu, S., Li, B., and Prasad, S. (1998). Reversed modeling of microwave circuits with bidirectional neural network models. *IEEE Transactions on Microwave Theory and Techniques*, 46(10):1492–1494.
- [Vapnik et al., 1997] Vapnik, V., Golowich, S., and Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, 9:281–287.
- [Vapnik, 1992] Vapnik, V.Ñ. (1992). Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems*, 4:831–838.

- [Verma, 1995] Verma, B. (1995). Handwritten hindi character recognition using multilayer perceptron and radial basis function neural networks. *IEEE International Conference on Neural Networks (ICNN 95)*, 4:2111–2115.
- [Wang et al., 1998] Wang, F., Devabhaktuni, V. K., and Zhang, Q. J. (1998). A hierarchical neural network approach to the development of a library of neural models for microwave design. *IEEE Transactions on Microwave Theory and Techniques*, 46(12):2391–2403.
- [Wang y Zhang, 1997] Wang, F. and Zhang, Q. J. (1997). Knowledge-based neural models for microwave design. *IEEE Transactions on Microwave Theory and Techniques*, 45(12):2333–2343.
- [Wang et al., 2006] Wang, Y., Yu, M., Kabir, H., and Zhang, Q.-J. (2006). Effective design of cross-coupled filter using neural networks and coupling matrix. *IEEE International Microwave Symposium Digest*, pages 1431–1434.
- [Watson et al., 1999] Watson, P. M., Cho, C., and Gupta, K. C. (1999). Electromagnetic-artificial neural network model for synthesis of physical dimensions for multilayer asymmetric coupled transmission structures. *International Journal of RF and Microwave CAE*, 9(3):175–186.
- [Watson y Gupta, 1997] Watson, P. M. and Gupta, K. C. (1997). Design and optimization of cpw circuits using em-ann models for cpw components. *IEEE Transactions on Microwave Theory and Techniques*, 45(12):2515–2523.
- [Weller, 2000] Weller, T. M. (2000). Edge-coupled coplanar waveguide bandpass filter design. *IEEE Transactions on Microwave Theory and Techniques*, 48(12):2453–2458.
- [Weng et al., 2007] Weng, W.-D., Yang, C.-S., and Lin, R.-C. (2007). A channel equalizer using reduced decision feedback chebyshev functional link artificial neural networks. *Information Sciences*, 177(13):2642–2654.
- [Widrow, 1962] Widrow, B. (1962). *Generalization and information storage in networks of adaline ‘neurons’ in Self-Organizing Systems*. Artech House.
- [Widrow y Hoff, 1960] Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON coinvention record*, pages 96–104.
- [Willshaw y der Malsburg, 1976] Willshaw, D. J. and der Malsburg, C. V. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London Series B*, 194:431–445.
- [Yang et al., 1992] Yang, J. J., Chow, Y. L., Howard, G. E., and Fang, D. G. (1992). Complex image of an electric dipole in homogeneous and layered dielectrics between two ground planes. *IEEE Transactions on Microwave Theory and Techniques*, 40(3):595–599.
- [Yang y Tseng, 1996] Yang, S.-S. and Tseng, C.-S. (1996). An orthogonal neural network for function approximation. *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, 26(5):779–785.
- [Yee, 1966] Yee, K. S. (1966). Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14:302–307.

- [Yuan et al., 1990] Yuan, X., Lynch, D. R., and Strohbehn, J. W. (1990). Coupling of finite element and moment methods for electromagnetic scattering from inhomogeneous objects. *IEEE Transactions on Microwave Theory and Techniques*, 38(3):386–393.
- [Zaabab et al., 1995] Zaabab, A. H., Zhang, Q. J., and Nakhla, M. S. (1995). A neural network modeling approach to circuit optimization and statistical design. *IEEE Transactions on Microwave Theory and Techniques*, 43(6):1349–1358.
- [Zhang et al., 2003] Zhang, Q. J., Gupta, K. C., and Devabhaktuni, V. K. (2003). Artificial neural networks for rf and microwave design—from theory to practice. *IEEE Transactions on Microwave Theory and Techniques*, 51(4):1339–1350.
- [Zhao y Atkeson, 1996] Zhao, Y. and Atkeson, C. G. (1996). Implementing projection pursuit learning. *IEEE Transactions on Neural Networks*, 7(2):362–373.

Índice de figuras

1.1. Estructura multicapa apantallada estudiada en este trabajo.	2
2.1. Dipolo elemental radiante insertado en una estructura multicapa apantallada y su modelo circuital equivalente.	27
2.2. Voltaje, intensidades e impedancias de entrada del modelo circuital equivalente. Para facilitar la comprensión del esquema se ha omitido la dependencia con el índice del modo (m, n) y el tipo de modo (TE,TM)	29
2.3. Red de impedancias equivalente de una línea de transmisión.	29
2.4. Serie de imágenes espaciales necesarias para satisfacer las condiciones de contorno en las paredes metálicas para una carga puntual en (x', y')	33
2.5. Representación del conjunto de capas de influencia.	34
2.6. Modelo circuital equivalente necesario para calcular el voltaje de la parte cuasi-estática de las funciones de Green de los potenciales.	35
2.7. Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado en el centro del plano XY.	40
2.8. Curvas de nivel correspondientes al número de capas de influencia necesarias para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	40
2.9. Curvas de nivel correspondientes al número de capas de influencia necesarias para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra cerca de una esquina de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	41
2.10. Orden del resto ($M=N$) necesario para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	48

2.11. Orden del resto ($M=N$) necesario para alcanzar una precisión de 10^{-5} cuando el dipolo excitación se encuentra cerca de una esquina de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	48
2.12. Tiempo de ejecución empleado para calcular las funciones de Green en función del umbral U_x cuando el punto fuente se encuentra situado en el centro de la cavidad de la Figura 2.7. Los puntos observación pertenecen a la recta $y/b = 0,50$ situada sobre el punto fuente. La frecuencia es 10 GHz.	49
2.13. Tiempo de ejecución empleado para calcular las funciones de Green en función del umbral U_x cuando el punto fuente se encuentra situado cerca de las paredes de la cavidad de la Figura 2.7. Los puntos observación pertenecen a la recta $y/b = 0,0960$ situada sobre el punto fuente. La frecuencia es 10 GHz.	49
2.14. Parte real de la función de Green del potencial eléctrico escalar G_{cV} correspondiente a un dipolo excitación situado en el centro de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	51
2.15. Parte real de la función de Green del potencial eléctrico escalar G_{cV} correspondiente a un dipolo excitación situado cerca de las paredes de la cavidad de la Figura 2.7. La frecuencia es 10 GHz.	51
2.16. Interfaz entre capas dieléctricas en una estructura multicapa apantallada. Se muestran cuatro cuadrantes en los que se divide de forma regular la cavidad. En cada cuadrante aparece un dipolo excitación y un punto observación.	52
3.1. Estructura de un Perceptrón multicapa. Las funciones φ de las capas ocultas pueden ser de igual o diferente tipo que las funciones de la capa de salida φ_s . El espacio de entrada posee dimensión n y el espacio de salida dimensión m	62
3.2. Estructura de una Máquina de vectores soporte (SVM). Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. La capa oculta realiza una transformación no lineal. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida.	64
3.3. Patrones en el espacio original y en el espacio de la capa oculta de la RBFNN.	66
3.4. Salida de la RBFNN en el espacio oculto (a) y en el espacio original (b).	66
3.5. Estructura de una RBFNN. Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. La capa oculta realiza una transformación no lineal. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida. En cada función neuronal $\phi(\mathbf{x}, \mathbf{c}_i) = \phi(\ \mathbf{x} - \mathbf{c}_i\)$. El término b es una constante que suele valer la unidad y está multiplicada por un conjunto de pesos adecuado.	70

3.6. Grupos de datos de entrada del conjunto de entrenamiento y centros seleccionados por el algoritmo de agrupación de K-medias.	73
3.7. Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado en el centro del plano transversal. . . .	83
3.8. Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en el centro de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.	83
3.9. Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en cerca de las paredes de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.	84
3.10. Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado en el centro de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.	84
3.11. Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7 cuando no existen tapas superior ni inferior.	85
3.12. Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado en el centro de la cavidad representada en la Figura 3.7.	85
3.13. Parte real de la función de Green del potencial eléctrico escalar. Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7.	86
3.14. Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado en el centro de la cavidad representada en la Figura 3.7.	86
3.15. Efecto de la extracción de la singularidad en la función de Green del potencial eléctrico escalar (parte real). Dipolo situado cerca de las paredes de la cavidad representada en la Figura 3.7.	87
3.16. Rejilla de puntos de entrenamiento y validación. Los puntos fuente de entrenamiento son asteriscos y los puntos observación son círculos. Los puntos fuente del conjunto de validación se representan con cuadrados y los puntos observación con signos suma.	88
3.17. Estructura de la RBFNN que aproxima las partes reales e imaginarias de las funciones de Green multiplicadas por (ρ)	89
3.18. Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de entrenamiento cuando la cavidad de la Figura 3.7 no posee tapa superior ni inferior.	92

3.19. Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de validación cuando la cavidad de la Figura 3.7 no posee tapa superior ni inferior.	93
3.20. Aproximación de la parte real de G_V una vez eliminada la singularidad para dos puntos fuente de la estructura 3.7 sin tapas superior ni inferior.	94
3.21. Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de entrenamiento en la cavidad de la Figura 3.7 apantallada totalmente.	95
3.22. Error NMSE en la aproximación de las funciones de Green (sin singularidad) del conjunto de validación en la cavidad de la Figura 3.7 apantallada totalmente.	95
3.23. Aproximación de la parte real de G_V una vez eliminada la singularidad para dos puntos fuente de la estructura 3.7 totalmente apantallada.	96
3.24. Error NMSE para un MLP diseñado con siete algoritmos de entrenamiento diferentes. La estructura de trabajo corresponde a la cavidad dibujada en la Figura 3.7 sin tapas superior ni inferior.	99
3.25. Error NMSE para un MLP diseñado con siete algoritmos de entrenamiento diferentes. La estructura de trabajo corresponde a la cavidad dibujada en la Figura 3.7 totalmente apantallada.	99
3.26. Error NMSE en una SVM para diferentes valores de varianza.	101
3.27. Número de vectores soporte para diferentes valores de varianza.	102
3.28. Ejemplo de zonas observación cuadradas. Los puntos observación se representan mediante puntos oscuros.	104
3.29. Ejemplo de zonas fuente. Los puntos fuente se representan mediante puntos transparentes.	104
3.30. Estructura de la RBFNN que aproxima las partes reales e imaginarias de las funciones de Green multiplicadas por (ρ) en un ancho de banda de trabajo. La coordenada frecuencial del dato de entrada se expresa en GHz. Cada función gaussiana ϕ posee un centro y una varianza.	105
3.31. Diagrama de flujo del algoritmo de división en zonas de la coordenada frecuencia. . .	107
3.32. Comparativa del rendimiento entre los algoritmos OLS y K-medias para la primera zona observación.	110
3.33. Estructura del filtro paso bajo impreso en el plano transversal de la cavidad apantallada multicapa de la Figura 3.7.	111
3.34. Parámetros S del filtro paso bajo de la Figura 3.33(a).	112

3.35. Diagrama de flujo del algoritmo de división en zonas de los puntos observación.	114
3.36. Diagrama de flujo del algoritmo de división en zonas de los puntos fuente.	116
3.37. Filtro paso banda de cuatro polos basado en líneas acopladas. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).	117
3.38. Filtro paso banda de cuatro polos. Las líneas acopladas se conectan con una longitud extra de línea de transmisión. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).	118
3.39. Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,20.	119
3.40. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,20.	119
3.41. Mallados de los filtros de cuatro polos mostrados en las Figuras 3.37 y 3.38.	120
3.42. Diagrama de flujo del algoritmo de división en zonas de los puntos observación cuando se aplica el mecanismo de varianza adaptativa.	123
3.43. Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. Para validar la técnica se añaden valores correspondientes a medidas. La respuesta se simula para 151 puntos de frecuencia.	125
3.44. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. Para validar la técnica se añaden valores correspondientes a medidas. La respuesta se simula para 151 puntos de frecuencia.	125
3.45. Filtro paso banda de seis polos. Se incluyen las dimensiones de la estructura multicapa apantallada (misma estructura de la Figura 2.7 de la Sección 2.3).	126
3.46. Respuesta del filtro de sexto orden de la Figura 3.45 para la solución directa y para el método neuronal entrenado con un error NMSE de 0,10 para la primera zona observación mediante el mecanismo de varianza adaptativa. La respuesta se simula para 151 puntos de frecuencia.	127
3.47. Estructura apantallada multicapa y dos dipolos. Un dipolo está situado cerca de las paredes metálicas y el otro dipolo está situado cerca del centro del plano transversal. Misma estructura de la Figura 2.7 de la Sección 2.3 y utilizada en los filtros 3.37,3.38 y 3.45.	131

3.48. Comparativa entre las dos técnicas de eliminación de la singularidad desarrolladas. . .	131
3.49. Efecto de la extracción de dos capas de influencia en la función de Green del potencial eléctrico escalar. Punto fuente situado cerca del centro de la cavidad representada en la Figura 3.47.	132
3.50. Efecto de la extracción de dos capas de influencia en la función de Green del potencial eléctrico escalar. Punto fuente situado cerca de las paredes de la cavidad representada en la Figura 3.47.	132
3.51. Ejemplos de zonas circulares y cuadradas. Los puntos negros simbolizan puntos observación.	133
3.52. Número de multiplicaciones para los algoritmos FOLS, OLS clásico y OLS mejorado para diferentes tamaños del conjunto de entrenamiento N	137
3.53. Zonas fuente y cavidades metálicas empleadas para testear las técnicas de aproximación de las funciones de Green.	138
3.54. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$	139
3.55. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$	139
3.56. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$	140
3.57. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$	141
3.58. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$	142
3.59. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$	143
3.60. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. . .	143

3.61. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente situada cerca de las paredes en las cavidades 1 y 2 de la Figura 3.53. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$	144
3.62. Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	150
3.63. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	150
3.64. Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.37. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua, cuadrados).	151
3.65. Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.38. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua, cuadrados).	152
3.66. Filtro paso banda de cinco polos basado en líneas acopladas. Se incluyen las dimensiones de la estructura multicapa apantallada (Cavidad 1 de la Figura 3.53).	152
3.67. Respuesta del filtro de quinto orden de la Figura 3.66 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). La respuesta se simula para 151 puntos de frecuencia.	153
3.68. Comparativa entre las respuestas del filtro de quinto orden de la Figura 3.66. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes).	153

3.69. Filtro de quinto orden de tipo horquilla. Se incluyen la estructura del filtro y las dimensiones de la caja (cavidad 2 de la Figura 3.53).	160
3.70. Respuesta del filtro de quinto orden de tipo horquilla de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con zonas circulares y la nueva técnica de extracción de la singularidad (línea de puntos y rayas). Para validar las simulaciones se exponen las medidas (línea punteada). La respuesta se simula para 151 puntos de frecuencia.	161
3.71. Fotografía del filtro de cinco polos de tipo horquilla fabricado.	161
3.72. Comparativa entre las respuestas del filtro de cuarto orden de la Figura 3.69. Los resultados se muestran para la solución directa (línea continua, cruces), para la nueva técnica de extracción con zonas circulares (línea con guiones, asteriscos), para la técnica de multiplicación por (ρ) combinada con zonas circulares (línea continua, círculos), y para la técnica de multiplicación por (ρ) con zonas cuadradas (línea con guiones, diamantes). Para validar las simulaciones se exponen las medidas (línea continua). . .	162
4.1. Los cinco primeros polinomios de Chebychev en el rango $[-1, 1]$	175
4.2. Estructura de una Red neuronal de Chebychev. Las neuronas de entrada y salida no realizan ninguna operación sobre los datos. Cada neurona de la capa oculta realiza una transformación no lineal definida por un polinomio de Chebychev particular. La capa de salida realiza una transformación lineal mediante los pesos que conectan las neuronas de la capa oculta y las neuronas de salida.	178
4.3. Función neuronal generada por la multiplicación del polinomio de Chebychev de primer y segundo orden (línea continua). Con guiones se dibuja la función neuronal obtenida con la multiplicación de los polinomios de orden 30 y 40.	182
4.4. Número de multiplicaciones para los algoritmos FOLS en una RBFNN (línea punteada), OLS con simplificación (línea con guiones) y sin simplificación (línea con punto y raya) en las Redes neuronales de Chebychev.	190
4.5. Funciones utilizadas en el test numérico de aproximación de funciones unidimensionales.	192
4.6. Error de tipo NMSE para las tres redes neuronales en la aproximación de la función de la ecuación (4.46).	193
4.7. Ratio de reducción del error en la aproximación de la función de la ecuación (4.46). Los ratios ordenados de mayor a menor correspondientes a la aplicación del algoritmo OLS se dibujan con cruces. Los ratios desordenados, correspondientes a polinomios de orden creciente (sin algoritmo OLS), se dibujan con círculos.	194
4.8. Ratio de reducción del error en la aproximación de la función de la ecuación (4.47). . .	195

4.9. Funciones utilizadas en el test numérico de aproximación de funciones multidimensionales.	197
4.10. Nuevas sub-zonas observación cuadradas generadas para el entrenamiento de las Redes neuronales de Chebychev.	201
4.11. Primera zona observación en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev.	202
4.12. Zona observación arriba y derecha en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev.	203
4.13. Primera zona fuente generada para el entrenamiento de las Redes neuronales de Chebychev. Los círculos son puntos de entrenamiento y los asteriscos son puntos de validación.	203
4.14. Nuevas sub-zonas fuente generadas para el entrenamiento de las Redes neuronales de Chebychev. Los círculos y los cuadrados son puntos de entrenamiento, los asteriscos y los cuadrados son puntos de validación.	204
4.15. Primera zona fuente, sub-zona fuente vertical y sub-zona fuente derecha en el espacio $[-1, 1]^2$ de trabajo de las Redes neuronales de Chebychev. Los círculos son puntos de entrenamiento, los asteriscos son puntos de validación.	205
4.16. Error cuadrático medio de los conjuntos de validación correspondientes a la zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. La cavidad corresponde a la estructura numerada con un 1 en la Figura 4.18. La frecuencia se fija en 11 GHz.	206
4.17. Error cuadrático medio de los conjuntos de validación correspondientes a la zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. La cavidad corresponde a la estructura numerada con un 2 en la Figura 4.18. La frecuencia se fija en 3 GHz.	206
4.18. Zonas fuente y cavidades metálicas empleadas para testear la capacidad de aproximación de las funciones de Green por parte de las Redes neuronales de Chebychev. Las cavidades 1 y 2 son las mismas que fueron utilizadas en el Capítulo 3 como se observa en la Figura 3.53. La diferencia entre esta figura y la Figura 3.53 estriba en que ahora las zonas fuentes tienen un mayor área para mostrar la capacidad de aproximación de las Redes neuronales de Chebychev.	207
4.19. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.	208

4.20. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.	208
4.21. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.	209
4.22. Error cuadrático medio de los conjuntos de validación correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.	210
4.23. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.	210
4.24. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente alejada de las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con círculos y los de la técnica de extracción de dos capas de imágenes con cruces.	211
4.25. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de mayor tamaño con un área igual a $7,9 \cdot 10^{-4}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.	211

4.26. Error cuadrático medio de los conjuntos de validación con la singularidad recuperada correspondientes a la zona fuente cercana a las paredes en las cavidades 1 y 2 de la Figura 4.18. Zona observación de menor tamaño con un área igual a $7,9 \cdot 10^{-6}$. Los resultados de la técnica de multiplicación por (ρ) se muestran con diamantes en la sub-zona vertical y con círculos en la sub-zona horizontal. Los resultados de la técnica de extracción de dos capas de imágenes se muestran con asteriscos en la sub-zona vertical y con cruces en la sub-zona horizontal.	212
4.27. Respuesta del filtro de cuarto orden de la Figura 3.37 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	219
4.28. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	219
4.29. Respuesta del filtro de quinto orden de la Figura 3.66 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). La respuesta se simula para 151 puntos de frecuencia.	220
4.30. Respuesta del filtro de quinto orden de tipo horquilla de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas). Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia. . . .	220
4.31. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas) cuando se consideran pérdidas en el dieléctrico de valor $\tan \delta = 0,003$. Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	222
4.32. Respuesta del filtro de quinto orden de la Figura 3.69 para la solución directa (línea continua) y para el método neuronal con Redes neuronales de Chebychev (línea de puntos y rayas) cuando se consideran pérdidas en el dieléctrico de valor $\tan \delta = 0,002$. Para validar la técnica se añaden valores correspondientes a medidas (línea de puntos). La respuesta se simula para 151 puntos de frecuencia.	222
4.33. Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de las Figuras 3.37 y 3.38 para la técnica definida en el dominio espectral. La respuesta se muestra para 25000 modos (línea con guiones), 50000 modos (línea continua), 75000 modos (línea con guión y puntos) y para 100000 modos (línea punteada).	224

4.34. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espectral. La respuesta se muestra para 25000 modos (línea con guiones), 50000 modos (línea continua), 75000 modos (línea con guión y puntos) y para 100000 modos (línea punteada).	224
4.35. Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de las Figuras 3.37 y 3.38 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 75000 modos (línea con guión y puntos) y el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).	225
4.36. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 75000 modos (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones).	225
4.37. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espectral. La respuesta se muestra para 1500 modos (línea con guiones), 2500 modos (línea continua), 5000 modos (línea con guión y puntos) y para 10000 modos (línea punteada).	226
4.38. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espacial (línea continua), la técnica definida en el dominio espectral con 5000 modos (línea con guión y puntos) y el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).	227
4.39. Módulo del coeficiente de reflexión S_{11} de los filtros de cuarto orden de las Figuras 3.37 y 3.38 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS [®] (línea con guión y puntos), y para el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).	227
4.40. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.66 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS [®] (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones).	228
4.41. Módulo del coeficiente de reflexión S_{11} del filtro de quinto orden de la Figura 3.69 para la técnica definida en el dominio espacial (línea continua), para la herramienta comercial ADS [®] (línea con guión y puntos), para el método neuronal de Chebychev (línea con guiones). También se muestran las medidas (línea punteada).	228
4.42. Número de análisis circuitales necesarios para convertir en rentable el método neuronal de Chebychev en un filtro típico impreso en la primera cavidad de la Figura 4.18. Se muestran los resultados para un valor medio de evaluación de las funciones de Green de 10^{-1} segundos (línea continua), 10^{-2} segundos (línea con guiones), 10^{-3} segundos (línea con guión y puntos), 10^{-4} segundos (línea punteada).	231

4.43. Número de análisis circuitales necesarios para convertir en rentable el método neuronal de Chebychev en un filtro típico impreso en la segunda cavidad de la Figura 4.18. Se muestran los resultados para un valor medio de evaluación de las funciones de Green de 10^{-1} segundos (línea continua), 10^{-2} segundos (línea con guiones), 10^{-3} segundos (línea con guión y puntos), 10^{-4} segundos (línea punteada)	232
4.44. Respuesta del filtro de cuarto orden de la Figura 3.38 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 101 puntos de frecuencia.	233
4.45. Filtro de segundo orden con anillos cuadrados abiertos. Se incluye la estructura del filtro y las dimensiones de la caja.	236
4.46. Mallado del filtro con anillos cuadrados abiertos.	236
4.47. Respuesta del filtro de segundo orden de la Figura 4.45 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 52 puntos de frecuencia.	237
4.48. Filtro de segundo orden con anillos cuadrados cerrados. Se incluye la estructura del filtro y las dimensiones de la caja.	237
4.49. Mallado del filtro con anillos cuadrados cerrados.	238
4.50. Respuesta del filtro de segundo orden de la Figura 4.48 para la solución directa (línea con círculos) y para el método neuronal con Redes neuronales de Chebychev para la caja variable (línea con cruces). La respuesta se simula para 52 puntos de frecuencia.	238
5.1. Estructura del cromosoma en el problema de diseño de filtros encapsulados. El cromosoma está compuesto por $n + m$ genes. Los n primeros genes codifican las separaciones. Los m últimos genes codifican las longitudes de los resonadores.	244
5.2. Ejecución del algoritmo de la ruleta en la selección de un individuo.	245
5.3. Puntos de cruce en un cromosoma compuesto por tres genes.	246
5.4. Ejemplo de dos funciones de rendimiento con diferentes valores de z (control de la pendiente). La función rendimiento correspondiente a la ecuación (5.2a) se dibuja con círculos. La función rendimiento correspondiente a la ecuación (5.2b) se dibuja con cruces.	247
5.5. Estructura y variables de un filtro de segundo orden formado por líneas acopladas. También se muestran las dimensiones y estructura de la cavidad.	250
5.6. Valores máximo y medio del rendimiento correspondiente al primer filtro de segundo orden. Los valores con línea continua son la media de cinco ejecuciones. Los valores sin línea continua son los extremos.	252

5.7. Valores máximo y medio del rendimiento correspondiente al segundo filtro de segundo orden. Los valores con línea continua son la media de cinco ejecuciones. Los valores sin línea continua son los extremos.	252
5.8. Respuesta de los filtros iniciales y optimizados de segundo orden. En línea continua se muestra la máscara de atenuación.	253
5.9. Ampliación de la banda de paso de la respuesta de los filtros optimizados de segundo orden. Para validar los resultados se muestran la solución directa obtenida con las funciones de Green evaluadas numéricamente para los dos filtros.	253
5.10. Estructura y variables de un filtro de cuarto orden formado por líneas acopladas. Se muestran las dimensiones de filtro inicial que debe ser optimizado. También se muestran las dimensiones y estructura de la cavidad.	256
5.11. Respuesta de un filtro ideal de tipo Chebychev de cuarto orden (línea continua) y respuesta de un filtro auxiliar (línea con guiones) para calcular los parámetros z de las funciones de rendimiento. El rizado en la banda de paso es 0,10 dB. Se muestra la máscara de atenuación con líneas continuas.	257
5.12. Funciones de rendimiento de los parámetros S_{11} y S_{21} correspondientes a los valores de parámetros z recogidos en la Tabla 5.2.	257
5.13. Valores máximo y medio del rendimiento correspondiente al filtro de cuarto orden. . .	259
5.14. La respuesta neuronal se dibuja con cruces, la solución directa obtenida con las funciones de Green evaluadas numéricamente se dibuja con círculos.	260
5.15. Respuesta del filtro optimizado de cuarto orden. La respuesta neuronal se dibuja con cruces, la solución directa obtenida con las funciones de Green evaluadas numéricamente se dibuja con círculos. En línea continua se muestra la máscara de atenuación. .	260
5.16. Características de la población inicial (primera generación) en el diseño del filtro de cuarto orden.	261
5.17. Estructuras cargadas de forma doble para el cálculo del factor de calidad externo y de los coeficientes de acoplo en resonadores de líneas acopladas.	264
5.18. Ejemplo del módulo del coeficiente de transmisión (S_{21}) empleado para calcular el factor de calidad externo cuando se emplea una estructura cargada de forma doble como la de la Figura 5.17(a).	264
5.19. Estructuras cargadas de forma simple para el cálculo del factor de calidad externo y de los coeficientes de acoplo en resonadores de tipo horquilla.	265

5.20. Ejemplo de fase del coeficiente de reflexión (S_{11}) (línea continua) y de su derivada (línea con guiones) empleada para calcular el factor de calidad externo y la frecuencia de resonancia cuando se utiliza una estructura cargada de forma simple como la de la Figura 5.19(a).	265
5.21. Ejemplo de módulo del coeficiente de transmisión (S_{21}) correspondiente a la estructura de la Figura 5.17(b) y de la Figura 5.19(b).	266
5.22. Esquema general de las redes neuronales utilizadas en la tarea de diseño. Hasta N redes neuronales pueden ser necesarias para calcular las dimensiones físicas deseadas.	267
5.23. Redes neuronales cuando se emplean bien resonadores de tipo líneas acopladas bien resonadores de tipo horquilla (valores entre paréntesis).	267
5.24. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.17(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	270
5.25. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.17(a).	270
5.26. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de líneas acopladas. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.17(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	271
5.27. Estructura multicapa apantallada (coincide con la estructura mostrada en la Figura 3.38) y filtro de cuarto orden.	271
5.28. Estructura multicapa apantallada (coincide con la estructura mostrada en la Figura 3.69) y filtro de cuarto orden.	272
5.29. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.19(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	273

5.30. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.19(a).	273
5.31. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de tipo horquilla. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.19(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos. . .	274
5.32. Solución inicial y optimizada del filtro de cuarto orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Se muestra la máscara de atenuación con líneas continuas.	276
5.33. Solución optimizada del filtro de cuarto orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral y la simulación con el programa ADS [®] . Se muestra la máscara de atenuación con líneas continuas.	276
5.34. Solución inicial y optimizada del filtro de segundo orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Se muestra la máscara de atenuación con líneas continuas.	277
5.35. Solución optimizada del filtro de segundo orden de líneas acopladas correspondiente a la caja de la Figura 5.27. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral y la simulación con el programa ADS [®] . Se muestra la máscara de atenuación con líneas continuas.	278
5.36. Solución inicial (línea con guión y puntos) y optimizada (línea continua) del filtro de quinto orden de tipo horquilla correspondiente a la caja de la Figura 5.28. Se muestra la máscara de atenuación con líneas continuas.	279
5.37. Fotografía del filtro de cinco polos de tipo horquilla fabricado.	279
5.38. Solución optimizada (línea continua) del filtro de quinto orden de tipo horquilla correspondiente a la Figura 5.28. Para validar las soluciones obtenidas se muestran la solución obtenida con la Ecuación integral (línea con guión y puntos) y las medidas (línea con guiones). Se muestra la máscara de atenuación con líneas continuas. . . .	280

5.39. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de líneas acopladas. Los factores de calidad han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.17(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	281
5.40. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de líneas acopladas. Las frecuencias de resonancia han sido calculadas mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.17(a).	282
5.41. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de líneas acopladas. Los coeficientes de acoplo han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.17(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	282
5.42. Respuesta del filtro inicial de cuarto orden de líneas acopladas correspondiente a las dimensiones de la Tabla 5.7. Asimismo, se muestra la máscara de atenuación.	284
5.43. Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de cuarto orden de tipo Chebychev realizado con líneas acopladas.	285
5.44. Respuesta del filtro optimizado de cuarto orden de tipo Chebychev de líneas acopladas. Asimismo, se muestra la máscara de atenuación. Para validar los resultados se muestra la solución directa (línea con guiones).	285
5.45. Respuesta del filtro inicial de tercer orden de tipo Butterworth de líneas acopladas correspondiente a las dimensiones de la Tabla 5.10. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}).	286
5.46. Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de tercer orden de Butterworth realizado con líneas acopladas.	287
5.47. Respuesta del filtro optimizado de tercer orden de tipo Butterworth de líneas acopladas. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}). Para validar los resultados se muestra la solución directa (línea con guiones).	288

5.48. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del factor de calidad externo cuando se emplean resonadores de tipo horquilla. Los factores de calidad han sido calculados mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el factor de calidad externo y las dimensiones físicas se muestra en la Figura 5.19(a). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	289
5.49. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo de la longitud del resonador cuando se emplean resonadores de tipo horquilla. Las frecuencias de resonancia han sido calculadas mediante el método neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre la longitud del resonador y las dimensiones físicas se muestra en la Figura 5.19(a).	289
5.50. Rendimiento de la red neuronal en el conjunto de validación en el problema del cálculo del coeficiente de acoplo cuando se emplean resonadores de tipo horquilla. Los coeficientes de acoplo han sido calculados mediante el métodos neuronal de análisis basado en Redes neuronales de Chebychev. La estructura usada para calcular las relaciones entre el coeficiente de acoplo y las dimensiones físicas se muestra en la Figura 5.19(b). Los valores de validación corresponden a la línea continua con cruces. La salidas de la RBFNN corresponden a los círculos.	290
5.51. Respuesta del filtro inicial de quinto orden formado con resonadores de tipo horquilla correspondiente a las dimensiones de la Tabla 5.6. Asimismo, se muestra la máscara de atenuación.	290
5.52. Respuesta de cuatro iteraciones del proceso de optimización directo del filtro de quinto orden de Chebychev realizado con resonadores de tipo horquilla.	291
5.53. Respuesta del filtro optimizado de quinto orden de Chebychev formado por resonadores de tipo horquilla. Asimismo, se muestra la máscara de atenuación. Para validar los resultados se muestra la solución directa (línea con guiones).	292
5.54. Respuesta del filtro inicial de cuarto orden de tipo Butterworth formado con resonadores de tipo horquilla correspondiente a las dimensiones de la Tabla 5.14. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}).	293
5.55. Respuesta de tres iteraciones del proceso de optimización directo del filtro de cuarto orden de Butterworth realizado con resonadores de tipo horquilla.	293
5.56. Respuesta del filtro optimizado de cuarto orden de tipo Butterworth formado por resonadores de tipo horquilla. Asimismo, se muestra la máscara de atenuación para el coeficiente de transmisión (S_{21}). Para validar los resultados se muestra la solución directa (línea con guiones).	294

Índice de tablas

2.1. Expresión de las funciones f y g para las diferentes funciones de Green de los potenciales.	26
2.2. Valor de las funciones signo empleadas para definir las funciones de Green apantalladas	32
3.1. Resultados en tiempo de cálculo para los algoritmos de entrenamiento estudiados. Todas las simulaciones se llevaron a cabo con un ordenador con un procesador Pentium IV a 3GHz y con 4GB de memoria RAM.	97
3.2. Niveles NMSE de entrenamiento y validación para la cavidad sin tapas superior ni inferior.	100
3.3. Niveles NMSE de entrenamiento y validación para la cavidad totalmente apantallada.	100
3.4. Zonas observación generadas.	108
3.5. Zonas fuente generadas para la primera zona observación.	108
3.6. Zonas frecuenciales generadas.	109
3.7. Comparación de los resultados temporales de la solución directa y el método neuronal para el cálculo de los parámetros S en 51 puntos de frecuencia.	113
3.8. Zonas observación generadas por el algoritmo de división con varianza fija.	118
3.9. Zonas fuente generadas por el algoritmo de división con varianza fija.	118
3.10. Zonas observación generadas por el algoritmo de división con varianza adaptativa. .	124
3.11. Zonas fuente generadas por el algoritmo de división de fuentes con la primera zona observación definida con el algoritmo de varianza adaptativa.	124
3.12. Tiempo invertido en la selección de las zonas observación, fuente, frecuencia y tiempo de entrenamiento de las RBFNNs cuando se emplea el algoritmo con varianza fija. . .	127
3.13. Tiempo invertido en la selección de las zonas observación, fuente, frecuencia y tiempo de entrenamiento de las RBFNNs cuando se emplea el algoritmo con varianza adaptativa.	128

3.14. Comparación de los resultados temporales de la solución directa y el método neuronal con varianza adaptativa para el cálculo de los parámetros S en 151 puntos de frecuencia.	128
3.15. Zonas observación circulares con extracción de dos capas situadas en la región MS definida como: $0,0 \leq (x - x' /a, y - y' /b) \leq 0,20/\pi$. Cavidad 1 de la Figura 3.53. . .	145
3.16. Zonas observación cuadradas con multiplicación por (ρ) en la región DS. Cavidad 1 de la Figura 3.53.	145
3.17. Zonas fuente correspondientes a las zonas observación circulares con extracción de dos capas de la región MS. Zona de transición de la región MS con multiplicación por (ρ) . Cavidad 1 de la Figura 3.53.	146
3.18. Zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) en la región DS. Cavidad 1 de la Figura 3.53.	147
3.19. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Nueva técnica de extracción y zonas circulares en la región MS. Multiplicación por (ρ) y zonas cuadradas en la región DS.	147
3.20. Zonas observación circulares con multiplicación por (ρ) situadas la zona MS definida como: $0,0 \leq (x - x' /a, y - y' /b) \leq 0,20/\pi$. Cavidad 1 de la Figura 3.53.	148
3.21. Tres primeras zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Las zonas fuente de la zona de transición son las mismas que las exhibidas en 3.17. Cavidad 1 de la Figura 3.53.	148
3.22. Tres últimas zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 1 de la Figura 3.53.	149
3.23. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Técnica de multiplicación por (ρ) y zonas circulares en la zona MS. Técnica de multiplicación por (ρ) y zonas cuadradas en la zona DS. . .	149
3.24. Comparación en la primera cavidad de los resultados temporales de la solución directa y los tres métodos neuronales para el cálculo de los parámetros S en 151 puntos de frecuencia.	155
3.25. Zonas observación circulares con extracción de dos capas situadas en la zona MS, definida como: $0,0 \leq (x - x' /a, y - y' /b) \leq 0,20/\pi$. Cavidad 2 de la Figura 3.53. . .	155
3.26. Zonas observación cuadradas con multiplicación por ρ en la zona DS. Cavidad 2 de la Figura 3.53.	155
3.27. Zonas fuente correspondientes a las zonas observación circulares con extracción de dos capas de la región MS incluyendo la zona de transición. Cavidad 2 de la Figura 3.53.	156

3.28. Tres primeras zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) de la región DS. Cavidad 2 de la Figura 3.53.	156
3.29. Tres últimas zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) de la región DS. Cavidad 2 de la Figura 3.53.	156
3.30. Zonas observación cuadradas con multiplicación por (ρ) situadas en la zona MS definida como: $0,0 \leq (x - x' /a, y - y' /b) \leq 0,20/\pi$. Cavidad 2 de la Figura 3.53.	157
3.31. Cinco primeras zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.	157
3.32. Tres últimas zonas fuente correspondientes a las zonas observación circulares con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.	158
3.33. Zonas observación cuadradas con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.	158
3.34. Zonas fuente correspondientes a las zonas observación cuadradas con multiplicación por (ρ) . Cavidad 2 de la Figura 3.53.	159
3.35. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Nueva técnica de extracción y zonas circulares en la zona MS. Multiplicación por (ρ) y zonas cuadradas en la zona DS.	159
3.36. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Técnica de multiplicación por (ρ) y zonas circulares en la zona MS. Técnica de multiplicación por (ρ) y zonas cuadradas en la zona DS.	159
3.37. Comparación de los resultados temporales de la solución directa y los tres métodos neuronales para el cálculo de los parámetros S en 151 puntos de frecuencia. Segunda cavidad.	162
4.1. Error cuadrático medio del conjunto de entrenamiento y validación para la Red neuronal de Chebychev en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].	195
4.2. Error cuadrático medio del conjunto de entrenamiento y validación para la Red neuronal de Chebychev sin algoritmo OLS en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].	195
4.3. Error cuadrático medio del conjunto de entrenamiento y validación para la RBFNN en la aproximación de la función unidimensional del ejemplo 4 de [Lee y Jeng, 1998].	196
4.4. NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada con el algoritmo OLS. En cada caso con datos de entrenamiento sin ruido.	198

4.5. NMSE del conjunto de validación para la red RBFNN entrenada en cada caso con datos de entrenamiento sin ruido.	198
4.6. NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada sin el algoritmo OLS. En cada caso se emplean datos de entrenamiento sin ruido.	199
4.7. NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada con el algoritmo OLS. En cada caso con datos de entrenamiento con ruido blanco gaussiano.	199
4.8. NMSE del conjunto de validación para la red RBFNN entrenada en cada caso con datos de entrenamiento con ruido blanco gaussiano.	199
4.9. NMSE del conjunto de validación para la Red neuronal de Chebychev entrenada sin el algoritmo OLS. En cada caso se emplean datos de entrenamiento con ruido blanco gaussiano.	200
4.10. Tiempos de entrenamiento en segundos. El primer término es el número de puntos de entrenamiento para Redes neuronales de Chebychev y RBFNNs. El segundo término es el tamaño del conjunto de validación para las Redes neuronales de Chebychev. Entre paréntesis se muestra el tamaño del conjunto de validación para la RBFNN.	213
4.11. Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,20/\pi$ en la primera cavidad. Técnica de eliminación de la singularidad mediante la extracción de dos capas.	215
4.12. Zonas observación generadas en la región DS en la primera cavidad. Técnica de eliminación de la singularidad mediante la multiplicación por (ρ)	216
4.13. Número de zonas fuente correspondiente a las zonas observación de la región MS en la primera cavidad.	216
4.14. Número de zonas fuente correspondiente a las zonas observación de la región DS en la primera cavidad.	216
4.15. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la primera cavidad. Técnica de extracción de dos capas en la región MS. Multiplicación por (ρ) en la región DS.	217
4.16. Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,20/\pi$ en la segunda cavidad. Técnica de eliminación de la singularidad mediante la extracción de dos capas.	217
4.17. Zonas observación generadas en la región DS en la segunda cavidad. Técnica de eliminación de la singularidad mediante la multiplicación por (ρ)	217
4.18. Número de zonas fuente correspondiente a las zonas observación de la región MS en la segunda cavidad.	218

4.19. Número de zonas fuente correspondiente a las zonas observación de la región DS en la segunda cavidad.	218
4.20. Tiempo de selección de las zonas observación, fuente y frecuencia y tiempo de entrenamiento para la segunda cavidad. Técnica de extracción de dos capas en la región MS. Multiplicación por (ρ) en la región DS.	218
4.21. Comparación de los resultados temporales de la solución directa y el método neuronal basado en Redes neuronales de Chebychev para el cálculo de los parámetros S en 151 puntos de frecuencia.	221
4.22. Comparación de los resultados temporales del método neuronal basado en Redes neuronales de Chebychev, de la técnica de análisis definida en el dominio espectral (75000 modos) y del programa comercial ADS [®] . Todos los resultados corresponden al tiempo por punto en frecuencia en segundos.	226
4.23. Zonas observación generadas en la región MS definida como $0,0 \leq (abs(x - x')/a, abs(y - y')/b) \leq 0,10/\pi$ en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$). Técnica de eliminación de la singularidad mediante la extracción de dos capas.	234
4.24. Zonas observación generadas en la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$). Técnica de eliminación de la singularidad mediante la multiplicación por (ρ)	234
4.25. Número de zonas fuente correspondiente a las zonas observación de la región MS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).	234
4.26. Número de zonas fuente correspondiente a las cuatro primeras zonas observación de la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).	235
4.27. Número de zonas fuente correspondiente a las cuatro últimas zonas observación de la región DS en la cavidad base de las cajas dibujadas en las Figuras 4.45 y 4.48 ($a \times b = 35,00mm \times 25,00mm$).	235
5.1. Dimensiones de los filtros iniciales y optimizados de segundo orden en milímetros. . .	250
5.2. Valor de los parámetros z para los filtros de la Figura 5.10.	256
5.3. Dimensiones del filtro inicial y optimizado de cuarto orden en milímetros. L_A permanece constante y es igual a 7,538 mm.	259
5.4. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27.	275

5.5. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de segundo orden de líneas acopladas de la caja de la Figura 5.27.	277
5.6. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial y al filtro optimizado. Filtro de quinto orden de tipo horquilla de la caja de la Figura 5.28.	278
5.7. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27.	283
5.8. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de segundo orden de líneas acopladas de la caja de la Figura 5.27.	283
5.9. Incrementos en mm. aplicados a las separaciones y longitudes iniciales en la optimización el filtro de cuarto orden de líneas acopladas de la caja de la Figura 5.27. . . .	284
5.10. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de tercer orden de tipo Butterworth de líneas acopladas de la caja de la Figura 5.27.	286
5.11. Incrementos en mm. aplicados a las separaciones y longitudes iniciales en la optimización el filtro de tercer orden de líneas acopladas de tipo Butterworth de la caja de la Figura 5.27.	287
5.12. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de quinto orden de tipo horquilla de la caja de la Figura 5.28.	288
5.13. Valor en mm. aplicado a la longitud de los resonadores (L), la distancia de "tapping" (t) y la separación (S_2) inicial en la optimización el filtro de quinto orden con resonadores de tipo horquilla de la caja de la Figura 5.28.	291
5.14. Factor de calidad, coeficientes de acoplo y longitudes correspondientes al filtro inicial. Filtro de cuarto orden de tipo horquilla de la caja de la Figura 5.28.	292
5.15. Valor en mm. aplicado a la longitud de los resonadores (L) y la distancia de "tapping" (t) en la optimización el filtro de cuarto orden con resonadores de tipo horquilla de la caja de la Figura 5.28.	292
B.1. Estructura de los datos de entrenamiento: primeros $p(n - 1)$ bloques de $p(n)$ elementos cada uno.	316
B.2. Estructura de los datos de entrenamiento: bloques correspondientes a la variación de las dos últimas coordenadas.	317
D.1. Zonas fuente correspondientes a la primera zona observación de la región MS.	335

D.2. Zonas fuente correspondientes a la segunda zona observación de la región MS.	335
D.3. Zonas fuente correspondientes a la tercera zona observación de la región MS.	336
D.4. Zonas fuente correspondientes a la cuarta zona observación de la región MS.	336
D.5. Zonas fuente correspondientes a la primera zona observación de la región DS.	337
D.6. Zonas fuente correspondientes a la segunda zona observación de la región DS.	337
D.7. Zonas fuente correspondientes a la tercera zona observación de la región DS.	337
D.8. Zonas fuente correspondientes a la cuarta zona observación de la región DS.	338
D.9. Zonas fuente correspondientes a la quinta zona observación de la región DS.	338
D.10. Zonas fuente correspondientes a la primera zona observación de la región MS.	338
D.11. Zonas fuente correspondientes a la segunda zona observación de la región MS.	338
D.12. Zonas fuente correspondientes a la tercera zona observación de la región MS.	339
D.13. Zonas fuente correspondientes a la cuarta zona observación de la región MS.	339
D.14. Zonas fuente correspondientes a la primera zona observación de la región DS.	339
D.15. Zonas fuente correspondientes a la segunda zona observación de la región DS.	339
D.16. Zonas fuente correspondientes a la tercera zona observación de la región DS.	340
D.17. Zonas fuente correspondientes a la cuarta zona observación de la región DS.	340
D.18. Zonas fuente correspondientes a la quinta zona observación de la región DS.	340
D.19. Zonas fuente correspondientes a la primera zona observación de la región MS.	340
D.20. Zonas fuente correspondientes a la segunda zona observación de la región MS.	341
D.21. Zonas fuente correspondientes a la primera zona observación de la región DS.	342
D.22. Zonas fuente correspondientes a la segunda zona observación de la región DS.	342
D.23. Zonas fuente correspondientes a la tercer zona observación de la región DS.	342
D.24. Zonas fuente correspondientes a la cuarta zona observación de la región DS.	343
D.25. Zonas fuente correspondientes a la quinta zona observación de la región DS.	343
D.26. Zonas fuente correspondientes a la sexta zona observación de la región DS.	343
D.27. Zonas fuente correspondientes a la séptima zona observación de la región DS.	343

D.28.Zonas fuente correspondientes a la octava zona observación de la región DS.	343
--	-----